7-2015

Lidar and Machine Learning Estimation of Hardwood Forest Biomass in Mountainous and Bottomland Environments

Bowei Xue
University of Arkansas, Fayetteville

Lidar and Machine Learning Estimation of Hardwood Forest Biomass in Mountainous and Bottomland Environments

A thesis submitted in partial fulfillment
of the requirements for the degree of
Master of Arts in Geography

By

Bowei Xue
Northwest University
Bachelor of Science in
Management of Resource and Environment and
Urban-Rural Planning, 2012

July 2015
University of Arkansas

The thesis is approved for recommendation to the Graduate Council

Dr. Jason A. Tullis
Thesis Director

Dr. Jackson Cothren
Committee Member

Dr. Xuan Shi
Committee Member

Abstract

Light detection and ranging (lidar) has been applied in various forest applications, such as to retrieve forest structural information, to build statistical models for identification of tree species, and to monitor forest growth. However, despite significant progress in these areas, the choice of regression approach and parameter tuning remains an ongoing critical question. This study focused on choosing the right spatial generalization level to transform lidar point clouds to 2D images which can be further processed by mature image processing and pattern recognition approaches. It also compared the prediction ability of popular machine learning algorithms applied to aboveground forest biomass estimation. A neighborhood technique was employed to calculate lidar-derived height metrics which were used as predictors to estimate forest total biomass at the image object (or segment) level. Three machine learning algorithms were tested to explore the relationship between the lidar-derived height metrics and biomass observed *in situ*. The height metrics were calculated as percentile heights and canopy coverage based on the lidar points falling within certain spatial extents (neighborhoods). The effect of neighborhood size was examined by developing regression models using Support Vector Machine (SVM), Cubist, and Random Forest on images created by applying 0.5, 2.5, 5, 10, and 15-meter neighborhood. Experiments were conducted in two study sites, the Ozark Mountains of Arkansas and the Trinity River Basin of Texas, with significantly different landscapes, hardwood tree species, and lidar point distributions. Regression models were constructed and evaluated with 10-fold cross validation. Results showed that optimal neighborhood configurations depend on the lidar data and regression techniques that are applied. The optimal model among all neighborhoods and algorithms achieved training accuracies of 0.988 and 0.990, and validation accuracies of 0.902 and 0.853 (adjusted R^2) at the two study sites respectively.

Acknowledgement

This study would not have been possible without the data generously provided by Dr. Jason A. Tullis and Dr. Anthony Filippi. Thanks to Dr. Tullis for teaching me Cubist and lidar-related techniques. Thanks to Dr. Cothren for introducing me to a set of powerful tools of R. Thanks to Dr. Shi for teaching me ArcObjects and various programming techniques.

Table of Contents

1 Introduction

1.1 Forest study with lidar data

Forest inventory is an accounting of trees over a well-defined land area (Scott & Gove, 2002). Tree population, tree volumes, species composition, additional growth, and forest structural variables are commonly measured and recorded by trained surveyors in sampling plots. Aboveground biomass is one growth parameter that can be directly measured with destructive methods (e.g., cutting, drying, and weighing) or can be indirectly deduced using allometric equations. The purpose of measuring forest above-ground biomass includes, but not limited to, monitoring forest growth and estimating forest carbon storage. Forests serve as a major carbon sink and it becomes ever critical in today to survey the carbon storage of forests over large areas.

In forestry, above-ground biomass (AGB) is defined as the oven-dried mass of the above-ground portion of tree groups (Bortolot & Wynne, 2005). Forest biomass is one key parameter directly related to forest growth and forest carbon storage. It can be derived from forest structural parameters or be directly measured using destructive methods. Diameter at breast height (DBH) has been found to be highly correlated with biomass, and has been adopted in allometric equations to predict forest biomass. Jenkins et al. developed consistent and generalizable biomass regression equations with meta-analysis aiming to "provide a consistent basis for evaluating forest biomass across regional boundaries" (Jenkins et al., 2003). The Jenkins model consists of ten species-based national scale equations for estimating total above-ground biomass from DBH in United States. While these equations "are simple and consistent in format across the nation, they may not be sufficiently accurate for mid- to fine- scale analyses" (Zhou & Hemstorm, 2009).

Extensive field-measurements are typically reliable at the cost of being very resource intensive (Van Aardt et al., 2006). Remote sensing provides alternative approaches for assessment on forest biophysical parameters. Remotely-sensed data are less expensive and can meet the accuracy requirements of forest inventory. Remote sensing products provide detailed information of target objects over large areas with limited data collecting time which makes them suitable to map biomass at fine scales over broad spatial extents (Clark et al., 2011). On the other hand, ground-based measurements are typically used as the target variable in training regression models with remote sensing products as predictors.

Two remote sensing products are commonly available for the purpose of forest inventory. One common approach has been optical remote sensing. Forest inventory parameters (e.g. tree species, diameter at breast height, canopy coverage, and leaf area index) are correlated with electromagnetic reflectance characteristics revealed by digital images of moderate to high spatial resolution. Time series optical image analysis, including forest land surface phenology, can further support discovery of spatial-temporal changes of forest growth. High spatial resolution digital images offer accurate locational information which can be used to identify individual trees. Vegetation indices derived from hyperspectral and multispectral images have been employed to predict leaf area index, tree species, canopy coverage and other forest parameters. Land use and land cover data derived from digital images are useful ancillary data for forest management. Previous studies showed that remote sensing imagery has limited ability in investigating forest structural and growth parameters but can significantly increase the accuracy of individual tree identification. This may stem from the insufficient ability of remote sensing imagery to survey sub-canopies.

In contrast to aerial imaging techniques, light detection and ranging (lidar) samples 3D structures of targets, and usually provides higher accuracy when applied for forest biomass estimation than digital images. The biomechanical and ecological links between forest biomass and the vertical structure of forest woody components make it possible to estimate biomass from a lidar point cloud. Previous studies have detected strong correlation between lidar metrics and forest above-ground biomass (e.g. Drake et al., 2003; Popescu et al., 2009; Clark et al., 2011). Lidar is well-suited for the purpose of investigating the vertical and horizontal structures of forests. By counting the elapsed time of each returned laser pulse and recording the plane coordinate of each return, lidar point clouds are considered as real 3D models of the forests under study. Also, the unique capability of lidar penetrating into tree crowns leads to its wide application in describing the crown structure.

The distribution of laser canopy heights is linearly correlated with the vertical distribution of leaf area (Magnussen & Boudewyn, 1998) and can thus be applied to estimate biomass over a range of forest types at the stand and plot level (Lim & Treitz, 2004). The distribution of leaf area can be transformed into the distribution of leaf mass using specific leaf weight ratios. Above ground biomass is highly correlated with component biomass which is highly correlated with leaf area and leaf mass. Therefore, lidar-derived metrics are able to predict the total biomass. Percentile heights, which directly correspond to percentiles of laser canopy heights, are one set of such metrics.

The well-known canopy height model (CHM), which has been widely used to indicate tree height distributions, is a special case of percentile heights. CHM is a single band surface model that in a canopy may be conceptualized as pixel values representing the 100^{th} percentile (or maximum) height of lidar points. Though other percentiles are not included in CHM, it has

shown satisfactory capability to describe the forest canopies and to locate individual stems. The significant correlation between tree height and tree crown width justifies the use of CHM for locating individual trees and estimating tree biomass and other structural parameters at either individual tree- or plot-level (Wynne & Bortolot, 2005; Popescu et al., 2004; Popescu et al., 2003). CHM also provides a way to correct the slope-induced bias (**Figure 1**).

Percentile heights are discrete summary statistics of the tree structure within certain spatial units. Whereas, pseudo-waveforms offer an alternative way to describe the structural information in the form of continuous curves. Thus, existing signal processing techniques can be applied for either comparing tree growth across different spatial units or extracting features for building regression models. Metrics derived from pseudo-waveforms are considered to be related to biomass and canopy coverage (Muss et al., 2011). A pseudo-waveform may provide a more stable description of vertical distribution of tree components within a certain spatial unit than the percentile heights since it reconstructs the laser height distribution as a continuous function. Height metrics derived from discrete returns can be regarded as an approximation to waveforms (Van Aardt et al., 2006).

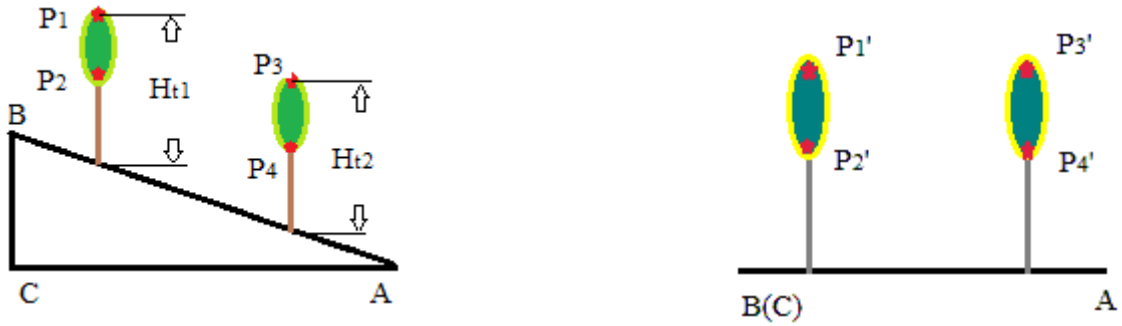


Figure 1 Slope-induced bias.

A lidar point cloud is regarded as an irregular sample of the target object. One common practice to exploit the structural information carried by lidar data is to generalize the original

data to images. Pixel values of such images are lidar-derived metrics designed for describing the distribution of lidar points within certain spatial units. Percentile height layers and canopy height models are examples of such generalized products. These images could be confidently used to examine the correlation between target forest biophysical variables and lidar-derived metrics and construct regression models at various scales.

To generate useful image products, the lidar data need to be classified. Lidar classification refers to assigning a unique classification code to each lidar point. The ASPRS LAS file format standards define 11 classes which have been widely adopted by lidar data processing softwares (**Table 1**). In this study, only points classified as high vegetation and ground were used to generate lidar height metric layers.

Table 1 ASPRS standard lidar point classes.

Classification Code	Classification Name
0	Created, never classified
1	Unclassified
2	Ground
3	Low Vegetation
4	Medium Vegetation
5	High Vegetation
6	Building
7	Low Point (noise)
8	Model Key-point (mass point)
9	Water
10	Reserved for ASPRS Definition
11	Reserved for ASPRS Definition
12	Overlap Points
13-31	Reserved for ASPRS Definition

(Adapted from ASPRS LAS 1.1/1.2 Format Standard)

With the classification information and return number, digital elevation models and canopy height models with various spatial resolutions could be derived. Using lidar point attributes and spatial query, lidar points falling within certain spatial units and complying with certain criteria can be extracted from the lidar point clouds. A set of statistics can be calculated to describe lidar points' z-value distributions including but not limited to maximum, minimum, mean, skewness, kurtosis, and ratios between points of different classification codes. Such statistics refer to lidar-derived metrics.

Applying the query-description process, height metric image layers can be derived. A neighborhood technique is one approach capable of producing lidar metric image layers of various spatial resolution and generalization extents. The technique requires three parameters to determine the image resolution and generalization extent: grid spacing, neighborhood shape and neighborhood size. For example, to produce a 1 by 1 m maximum height layer, the grid spacing and neighborhood size should be set as 1 meter. The neighborhood shape should be square.

Then, spatial queries are performed to find the maximum z-value within each 1 m^2 cell. This study used such neighborhood technique to derive lidar height metric layers from classified lidar point clouds.

Biomass estimations have been conducted at scales of pixel level, plot level, segment level and individual tree level. It might be argued that the individual tree level offers the most accurate and physically meaningful results. Forest biophysical parameters estimated at this level can be easily aggregated to larger scales. However, identifying individual trees brings additional errors and requires additional reference data. Accuracy of individual tree identification is constrained by the accuracy of positioning techniques, the availability of reference data, the performance of tree-finding algorithms, and the forest structure under study.

The segment level estimation is an efficient alternative to the individual tree level. The purpose of image segmentation is to group individual pixels into image objects that correspond to real objects. This technique has been applied on images of various scales from small-size photographs to satellite images covering large spatial extents. Several algorithms have been developed to find objects from given image layers. The general idea is to group adjacent pixels with similar attributes (color, texture, contextual information and other image features) into objects. Thus, image segmentation is different from unsupervised image classification which does not take into account spatial correlation. One specific algorithm being widely used for segmenting canopy height models is the multi-resolution image segmentation implemented by the commercial software eCognition. This algorithm employs a bottom-up process to connect pixels according to their spectral and geometric similarity.

Three dimensional viewing techniques allow for rendering lidar points directly or constructing 3D objects and animations using lidar data. For example, modelling individual trees

in forests. On the other hand, lidar points are traditionally used to produce surface images (e.g. digital elevation model, percentile height images) whose pixels are elevations or height values. The mean of pixel values in a plot is taken as an attribute for the plot. A third way is to identify individual trees, calculate height metrics for each tree, average over all trees then assign the result as a lidar height metric for that plot. Plots are then compiled as cases to build and test regression models. Lidar height metrics can also be calculated for each pixel. In this situation, a pixel needs to be large enough to encapsulate enough lidar points for calculating height metrics, but should also be small enough to capture spatial variation. As lidar technique advances, data with very high point densities become available. Lidar height metric images with fine spatial resolution can be used as inputs for image segmentation. Dense points also allow for identifying individual trees with high accuracy.

Plot size commonly varies from 300 m^2 to 900 m^2. A plot is regarded as a sample of the entire study site. Due to the cost of forest field survey, the number of plots is always limited. Thus, knowledge on the study site is required to make unbiased sampling. Additionally, it is common to include plots with no trees for AGB estimation as a baseline. This study incorporates two study sites with different dominant tree species and landscapes. Field measurements were conducted in a total of 45 forest inventory plots.

Airborne lidar data traditionally cover much smaller area compared to satellite images. The limited spatial coverage results in limited study area. Thus, most forest studies using lidar data focused on local variations. Regression models could only be reliably applied on forests with similar growth environment, tree species and tree age. Thus, the models are limited to local forests.

Though regression models are hard to be applied across study sites, the methods used to process lidar data and train models are exchangeable. Previous local efforts have constructed a big set of processing and computing components that could be recombined to fit the need of new local studies. Moreover, lidar data availability is expanding rapidly in terms of both decreased prices and enlarged data coverage.

1.2 Study objectives

This study employed three popular machine learning techniques to build regression models for estimating forest above-ground biomass at two study sites: the Ozark National Forest (ONF) and the Trinity River (TR). The whole process include (1) calculate lidar height metrics, (2) identify homogeneous forest units by image segmentation, and (3) use image segments as training data for building regression models.

Parameters for image segmentation, machine learning algorithms and the neighborhood technique need to be specifically tuned each time they are applied. These parameters involve neighborhood shape and size and grid spacing for the neighborhood technique, the C and sigma parameter for SVM, the number of committees and neighbors for Cubist regress tree, and number of variables randomly sampled at each split for Random Forest. More details about parameter tuning and model construction are discussed in section 3.3.4.

This study assumed that the field data is representative of the study sites. By examining model performance under different situation (different parameter combinations), this study explored the process of extracting useful structural information from lidar point could. The mechanism between parameter selection and predicting accuracy was also examined. In addition to estimating biomass for the two study sites, this study aimed to provide a data-driven approach with minimum human intervention on applying lidar in forest studies.

Study objectives include (1) compare the performance of biomass estimation models built with different algorithms and inputs; (2) select the most feasible neighborhood size for each study site and explore the influence of this parameter; and (3) construct an automated data-driven approach to biomass for utilizing lidar data.

2. Literature Review

2.1 Image segmentation

Image segmentation refers to the process of grouping image pixels. Each group is expected to represent a meaningful object. Some statistics, referred to as object attributes, are calculated for each group. Usually, the mean of pixel values are taken for further analysis. Segmentation is commonly regarded as the pre-step for image classification. Object-level image classification, unlike individual pixel based classification, accounts for spatial auto-correlation and image texture. Thus, pixels belonging to the same object but occupy very different digital numbers are more likely to be correctly classified.

This study adopted the object-based method to detect homogeneous forest unit within each plot. Homogeneous units are defined as spaces occupying the same attributes including DBH, canopy cover, height, crown closure and biomass. The segmentation process automatically detected such homogeneous units by maximizing between group variances and minizing within group variances. This optimization process differs from unsupervised image classification algorithms by the constraint that only spatially connected pixels get grouped.

Multiresolution image segmentation is a bottom-up region growing algorithm developed by Baatz & Schäpe (2000) and implemented by the commercial software eCognition. The algorithm calculates a heterogeneity measure for each object starting with single pixels and

iteratively merges objects while minimizing heterogeneity gain. At each step two objects are grouped into one larger object. The heterogeneity measure increases at each merge.

Among all adjacent objects of an object A, the algorithms searches for the object B which brings the least heterogeneity gain when merged with A. The algorithm then examines all adjacent objects of B and selects object C to merge with B. A and B will be actually merged if C equals to A. The heterogeneity gain can be viewed as a cost of merging. It grows from 0 to very large values as the algorithm iteratively merges objects. When the cost of merging becomes larger than a pre-defined threshold, the scale parameter, the corresponding merge will not be made. Thus, the threshold determines the largest size of objects and also the possible number of merges.

Baatz and Schäpe also defined the heterogeneity measure and cost of merging which they named the degree of fitting. The heterogeneity measure of one object consists of two elements: spectral heterogeneity and form heterogeneity.

For a single band, spectral heterogeneity equals to the standard deviation of pixel values. Let h_1 denote the single band heterogeneity of object A, h_2 denote the single band heterogeneity of object B, h_m denote the single band heterogeneity of the merged object, n_1 denote the number of pixels in object A, n_2 denote the number of pixels in object B, and h_{diff} denote the cost of fitting. Thus,

$$h_{diff} = h_m - \frac{h_1 * n_1 + h_2 * n_2}{n_1 + n_2}$$

or equivalently,

$$h_{diff} = (n_1 + n_2) * h_m - (n_1 * h_1 + n_2 * h_2) = n_1 * (h_m - h_1) + n_2 * (h_m - h_2)$$

This definition can be extended to any number of channels c, each with a weight w_c:

$$h_{diff} = \sum_{c} w_c(n_1 * (h_{mc} - h_{1c}) + n_2 * (h_{mc} - h_{2c}))$$

Object size is linearly correlated with the number of pixels it contains. The merged object contains exactly $(n_1 + n_2)$ pixels. This is compatible with the fact that objects are sequential subset.

Form heterogeneity consists of two elements: compactness and smoothness. Object compactness $h_{compact}$ is defined as $\frac{l}{\sqrt{n}}$, where l dotes the factual length of an object and n is the object size in pixels. Object smoothness h_{smooth} is calculated as $\frac{l}{b}$, where b is the shortest possible edge length given by the bounding box b of the object (Baatz & Schape). It is worth noting that this algorithm can be easily extended by using additional heterogeneity measures.

2.2 Regression algorithms

Relationships between real-world objects are the basis of estimating unknown situations or unobservable scenarios. Forest total biomass can be most accurately measured by field survey and lab analysis. However, the time and labor cost of this method is high. Additionally, this method limits the sample size and is not feasible for collecting data in unreachable zones. Most remote sensing techniques collect information (e.g. backscattered electromagnetic radiations) of objects inside interested areas with large spatial coverage rapidly. After careful processing, the remotely sensed information are transformed to variables (e.g. band reflectivity, coordinates, terrain heights, distances) stored in tables or data frames which could be used as inputs to various data mining tools.

Data mining techniques, ranging from ordinary least squares regression to artificial neuro networks have been closely combined with remotely sensed data to explore relationships between objects and to predict future trends. Forest total biomass has been proved to be

correlated with diameter at breast height, crown coverage and tree height. These relationships are determined by the biophysical characteristics of trees. Since forest physical structures are much easier to be measured by current remote sensing techniques, they have been widely used to estimate chemical attributes which are difficult or impossible to be measured without lab analysis.

It is desirable to select appropriate data mining tools to build regression models for biomass estimation. Machine learning provides a set of algorithms that are feasible for detecting data structures and building predictive models since they do not rely on data distribution assumptions. Three supervised learning algorithms were employed in this study: regression tree, random forest and support vector machine. In this section, a brief summary of the three data mining techniques is included.

2.2.1 Cubist

Decision trees represent a big family of tree structured classifiers and prediction tools which share the same principle. Many of them can be applied on both classification and regression problems. When the target variable is continuous, regression trees are built. When discrete class labels are to be predicted, classification trees are built. The input data for model training and testing are usually structured data frames with one target variable and several predictors. Most tree-structured classifiers or regression models are built using the same general procedure: growing a tree to its maximum size, pruning the tree, then testing its performance. Every group of tree-structured classifiers occupies a unique splitting rule, impurity measure, and pruning strategy.

Cubist is a regression tree algorithm developed by RuleQuest Research. It is an extension of Quinlan's M5 rule-based model (Max Kuhn et al., 2012). It was designed specifically for regression problems. Its counterpart for classification problems is the classification tree C5 which was also developed by RuleQuest Research.

M5 was proposed in Quinlan (1992) as a constructor of "tree-based piecewise linear models" (Quinlan 1992). Each leaf of a tree built by M5 can hold either a value or a multivariate linear model. Models trees constructed using M5 consist of a certain number of linear functions specifically fitted for a relatively small partition of the original data. M5 trees learn efficiently from data with up to hundreds of attributes and produce more accurate predictions. Input data for M5 is a data frame with user-specified attributes. Each row is a pair of predicting variables and one target variable. The prediction variables can be either discrete or continuous.

When constructing a model tree, M5 firstly grows a maximum tree then prunes it back for lowest generalization error. Let $(\mathbf{x_i}, y_i)$, i=1,2,3…,m denote input data, where $\mathbf{x}_i$ is a data vector contacting predicting values of case i, y_i is the associated target value. The impurity measure or error measure for regression trees, used by M5 is the standard deviation. Therefore, node t's impurity measure is calculated by

$$\mathrm{Sd(t)} = \sqrt{\frac{1}{N(t)} \sum_{x_n \in \mathrm{t}} (y_n - \bar{y}_t)^2}$$

A tree is initialized as a single root node containing all training data. Some tests are generated to affiliate each case to a subset according to the test results. The tests are splitting rules. Each splitting rule is a single question examining whether a predicting value of a case belongs to some subsets. All possible tests are applied, and the one generating the maximum impurity decline is used to actually split the node. For a node t, impurity decline is computed as:

$$\Delta \mathrm{Sd}(\mathrm{t}) = \mathrm{Sd}(\mathrm{t}) - \sum_{\mathrm{i}} \mathrm{P}(\mathrm{t}_\mathrm{i}) * \mathrm{Sd}(\mathrm{t}_\mathrm{i})$$

$$\mathrm{P}(\mathrm{t}_\mathrm{i}) = \frac{|\mathrm{t}_\mathrm{i}|}{|\mathrm{t}|}$$

where t_i are descendent nodes of t, and $|\mathrm{t}|$ is the number of training cases in node t. The splitting process is recursively performed on each node, until certain terminate conditions are satisfied.

All cases within a node are used to fit a multivariate linear model using attributes that appear in the splitting rules or be used to fit any linear models in the subtree of this node. M5 then compares the accuracy of the linear model of a node with the accuracy of the node's subtree. That is, a node's predicting performance is compared with its subtree. The accuracy of a linear model is measured by multiplying the mean of its predicting residue with $(\mathrm{n} + \mathrm{v})/(\mathrm{n} - \mathrm{v})$ in which n denotes the number of training cases and v denotes the number of model parameters. M5 also eliminates model parameters to minimize training error. The algorithm searches for the attributes (predicting variables) with little contribution to a linear model. In some cases, all variables are removed leaving only a constant (Quinlan 1992). Once the initial maximum tree has been constructed, M5 starts to prune beginning at the near-bottom nodes. Each non-leaf node is examined to see if its predicting accuracy is larger than its subtree. If so, its subtree is pruned and the non-leaf node is turned to a leaf node, otherwise, the subtree is kept.

M5 adopts a smoothing technique to improve predicting accuracy. Consider a novel case falling within a leaf node t. The path linking t with the root node t_0 is S. Let n_i denote the number of training cases in t, $\mathrm{PV}(\mathrm{t})$ denote the predicted value at the leaf node t, and $\mathrm{M}(\mathrm{t}_\mathrm{i})$

denote the predicted value given by the linear model inside an intermediate node t_i on S. Then the smoothed predicted value for this novel case is:

$$PV = \frac{n_i \times PV(t) + k \times M(t_i)}{n_i + k}$$

where k is a smoothing constant (Quinlan 1992).

Like M5, Cubist also generates a linear multivariate model or a constant value for each leaf node. It partitions the input data and fits a local linear model for each data subset instead of producing a general linear model for all the data. Therefore, the data partitioning carried by the splitting rules makes the two algorithms able to handle non-linear relationships without priori assumptions of data distributions. Quinlan additionally proposed a method to combine instance-based and model-based learning for better model performances. This technique can be applied on various algorithms which aim to build regression models. Both Cubist and M5 adopt the technique. In instance-based learning, a set of prototypes are generated to represent the training data. A prototype may be one of the training cases, or it may be a hypothetical case computed from several training cases (Quinlan 1993). A novel case is classified or predicted by finding its similar prototypes and use their target values in some way. On the other hand, explicit generalizations of the training cases are constructed in model-based learning (Quinlan 1993).

For a training set $(\mathbf{x}_i, y_i)$, the model-based approach constructs a model M while the instance-based approach generates a number of prototypes $\mathbf{p}_i$. For an unseen case **z**, M(**z**) is the predicted value given by the model. The similarities between **z** and the prototypes are also measured and prototype values on the target variable ($V(\mathbf{p}_i)$) are combined in some way to generate a predicted value for **z**. The model M can be used to predict the difference between target values of prototypes and **z**:

$$M(\mathbf{p}_i) - M(\mathbf{z})$$

then the target value of a prototype can be adjusted using the above calculation result:

$$V(\mathbf{p}_i) - (M(\mathbf{p}_i) - M(\mathbf{z}))$$

the two approaches are combined in such way and the adjusted prototype values are considered to be better predictors than the original values.

When composite models are determined to be sued in Cubist, the algorithm finds the n training cases that are closet to each novel case. Then, it calculates model predictions for the novel case **z** and its nearest prototypes: $M(\mathbf{z})$, $M(\mathbf{p}_i)$, i=1,2,…,n. The difference between the predicted target values of **z** and $\mathbf{p}_i$ is $M(\mathbf{z}) - M(\mathbf{p}_i)$. The adjusted prototype target values, $V(\mathbf{p}_i) + M(\mathbf{z}) - M(\mathbf{p}_i)$, are then averaged to give a prediction on **z**. The model M used by Cubist is the Cubist regression tree. Cubist can also implement committee models which consist of a set of regression trees. Each committee member predicts the target value for a case and their predictions are averaged to produce the final prediction. The first committee model equals to the model generated without using the committee technique. Succeeding models all attempt to compensate for the prediction errors of previous models. The number of nearest training cases and committee members are two parameters need to be tuned when building regression trees using Cubist.

2.2.2 Random forest

Random Forest is an expansion to the regression tree techniques. A combination of regression tree predictors are developed in a random "forest", where each tree is built on a random sample (with replacement) of observations. The following discussions on Random Forest are cited and summarized from the online material maintained by Breiman and Cutler.

The training sample used for building a single tree contains N cases (N equals to the number of total training cases). Pruning is not adopted, thus, each tree grows to its maximum size. A subset of features is also randomly selected for each tree. Usually, the number of features selected at a tree equals to log(M)+1, where M is the total number of features. Single tree's performance influences the overall performance of the forest. Besides, the classifier's error rate increases as correlations between individual trees increase. Approximately one third of the training sample is extracted to form the "OOB data" (out-of-bag data) at each individual classifier. OOB data are not used to train single tree classifiers, instead, they are used as validation sets to get unbiased estimates of single trees' classification errors, thus, no separate testing sets or cross validation is needed. For the entire forest, a validation classification is generated for each case on about one third of the trees. The class with highest vote every time a case is in OOB is taken as the predicted value for that case. The rates of misclassification over all cases are averaged to get the OOB error estimate.

The OOB data are also used to calculate variable importance. Random Forest permutes the values of variable m for each tree's OOB data and runs the single tree classifier on the altered data. The resultant predicting accuracy is subtracted from the predicting accuracy of the untouched OOB data. The average of this difference over all trees in the forest is the raw importance score of m. Variable importance enables feature selection in Random Forest. The whole classifier can be updated by running the algorithm again using variables with high importance only.

Random Forest also generates an N×N matrix each time it runs. The matrix measures the proximity of each pair of data in the training set. After a single tree classifier has been constructed, all data associated with this tree (both training and OOB cases) are classified by the

tree. The proximity of two cases is increased by one when they fall into the same leaf node of this tree. This procedure is applied on all trees, and the proximities are normalized by dividing by the number of trees. Proximity matrices are used to solve missing data problems and locating outliers.

Each tree only separates the observations with limited dimensions (features), but the forest as a whole performs quite well on the entire feature space. Random forest is a high-precision classifier with the ability to handle large inputs and to evaluate feature importance. It produces stable classification or regression results even with missing values in the inputs. Since only randomly selected cases and features are used to build each tree, the classic over-fitting problem is limited. Besides, Random Forest is fast. When being applied to regression problems, the predictions of a random forest are equal to the average of the predictions given by all trees inside the forest.

2.2.3 Support vector machine

Support vector machine is a statistical learning technique originally designed for binary classification. It learns from the training data and attempts to make correct predictions on novel data. It contains variants focusing on binary classification, outlier detection, or regression. As a supervised data mining tool, training data are required to build classifiers based on SVM. Input data for SVMs are compiled as data frame with columns as attributes and rows as cases. Each case could be viewed as an input vector **x** paired with a class label. SVM learns well from data showing nonlinear relationships between the target variable and predicting variables. The equations below are adapted from Colin et al. (2011).

For binary classification, the two classes are usually labeled by +1 and -1. The training data could be viewed as labeled data points in input space. SVM aims to find a hyper plane such that all data points labeled by +1 would be on one side of the plane and the other on the other side. The hyper plane should also occupy the maximum distance from the two classes of labeled points. The closest points on both sides of the plane have most influence on the position of this separating hyper plane. These points are called support vectors. The separating hyper plane could be written as

$$\mathbf{w} \cdot \mathbf{x} + \mathrm{b} = 0$$

where b is the offset of the hyper plane from the origin in input space, **x** are points located within the hyper plane, and **w** is the normal to the hyper plane.

The input training data set for a binary classification task involves m input vectors $\boldsymbol{x}_i$, each is a case or data point having corresponding class labels y_i=$\pm$1. Define a decision function

$$\mathrm{f(x)} = \mathrm{sign}(\mathbf{w} \cdot \mathbf{x} + \mathrm{b})$$

The decision function is invariant under any positive rescaling of the arguments inside the sign function. Thus, implicitly define a scale for (**w,**b) by setting $\mathbf{w} \cdot \mathbf{x} + \mathrm{b} = 1$ for the closest points on one side and $\mathbf{w} \cdot \mathbf{x} + \mathrm{b} = -1$ for the closest on the other side. The hyper planes passing through $\mathbf{w} \cdot \mathbf{x} + \mathrm{b} = 1$ and $\mathbf{w} \cdot \mathbf{x} + \mathrm{b} = -1$ are called canonical hyper planes, and the region between them is called the margin band. The distance between the two canonical hyper planes is

$$\frac{2}{\|\mathbf{w}\|_2}$$

define margin as

$$\gamma = \frac{1}{\|\mathbf{w}\|_2}$$

which equals to half of the distance between canonical planes. The generalization error bound on unseen cases could be minimized by maximizing the margin γ, the minimal distance between the hyper plane separating the two classes and the closest data points to the hyper plane. Thus, the basic objective of SVM is to minimize

$$\frac{\|\boldsymbol{w}\|_2^2}{2}$$

subject to

$$\mathrm{y_i}(\mathbf{w} \cdot \mathbf{x_i} + \mathrm{b}) \geq 1, \forall \mathrm{i}$$

The above formulation can be reduced to minimization of the primal Lagrange function:

$$\mathrm{L}(\mathbf{w}, \mathrm{b}) = \frac{1}{2}(\boldsymbol{w} \cdot \boldsymbol{w}) - \sum_{i=1}^{m} \alpha_i (y_i(\mathbf{w} \cdot \boldsymbol{x}_i + \mathrm{b}) - 1)$$

α_i are Lagrange multipliers, $\alpha_i \geq 0$. Take the derivatives with respect to b and **w**:

$$\frac{\partial L}{\partial b} = -\sum_{i=1}^{m} \alpha_i y_i = 0$$

$$\frac{\partial L}{\partial \boldsymbol{w}} = \boldsymbol{w} - \sum_{i=1}^{m} \alpha_i y_i \boldsymbol{x}_i = 0$$

solving **w** and substituting it back into L(**w**,b) results in the dual objective function:

$$\mathrm{W}(\boldsymbol{\alpha}) = \sum_{i=1}^{m} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{m} \alpha_i \alpha_j \, y_i y_j (\boldsymbol{x}_i \cdot \boldsymbol{x}_i)$$

maximizing $\mathrm{W}(\boldsymbol{\alpha})$ with respect to the constraints $\alpha_i \geq 0, \sum_{i=1}^{m} \alpha_i y_i = 0$ equals to minimizing L(**w**,b).

The input data may not be linearly separable. SVM uses a kernel function to map the original data into a feature space of higher dimensionality. In the mapped space, the data become linearly separable, and a hyper plane could be found to separate the two classes. The

generalization error bound would not be affected by the dimensionality of the space. The mapping function which maps data points to the new space is implicitly defined in

$$\mathbf{x}_\mathrm{i} \cdot \mathbf{x}_\mathrm{j} \rightarrow \Phi(\mathbf{x}_\mathrm{i}) \cdot \Phi(\mathbf{x}_\mathrm{j})$$

the kernel function $\mathrm{K}(\mathbf{x}_\mathrm{i}, \mathbf{x}_\mathrm{j}) = \Phi(\mathbf{x}_\mathrm{i}) \cdot \Phi(\mathbf{x}_\mathrm{j})$ determines that the feature space must be an inner product space. After a kernel function has been selected, the learning process involves maximization of the object function:

$$\mathrm{W}(\boldsymbol{\alpha}) = \sum_{\mathrm{i}=1}^{\mathrm{m}} \alpha_\mathrm{i} - \frac{1}{2} \sum_{\mathrm{i,j}=1}^{\mathrm{m}} \alpha_\mathrm{i} \alpha_\mathrm{j}\, \mathrm{y_i y_j K}(\mathbf{x}_\mathrm{i}, \mathbf{x}_\mathrm{i})$$

subject to

$$\alpha_i \geq 0, \forall \mathrm{i}$$

$$\sum_{i=1}^{m} \alpha_i y_i = 0$$

The offset b is calculated by solving the equation:

$$\mathrm{b} = -\frac{1}{2} \left(\max_{i|y_i=-1} \left(\sum_{j=1}^{m} \alpha_j y_j \mathrm{K}(\mathbf{x}_\mathrm{i}, \mathbf{x}_\mathrm{i}) \right) + \max_{i|y_i=+1} \left(\sum_{j=1}^{m} \alpha_j y_j \mathrm{K}(\mathbf{x}_\mathrm{i}, \mathbf{x}_\mathrm{i}) \right) \right)$$

Thus, to construct a binary classifier, SVM puts training data into $\mathrm{W}(\alpha)$ and solve the optimization problem. The optimum offset b^* is calculated from the optimum Lagrange multipliers α_i^* using the equation above. The optimum **w** is calculated by

$$\mathbf{w}^* = \sum_{i=1}^{m} \alpha_i^* y_i\, \Phi(\mathbf{x}_\mathrm{i})$$

Thus, for an input vector **z** outside the training set, its class is predicted based on the sign of the function:

$$\mathrm{u} = \sum_{\mathrm{i}=1}^{\mathrm{m}} \alpha_\mathrm{i}^*\, \mathrm{y_i K}(\mathbf{x}_\mathrm{i}, \mathbf{z}) + \mathrm{b}^*$$

$\mathbf{w}^*$ and $\mathbf{b}^*$ determine a separating hyper plane. Data points lying closest to this plane are support vectors. These data points have $\alpha_i^* > 0$ while other points have $\alpha_i^* = 0$. Therefore, the separating hyperplane and the decision function are only influenced by the support vectors.

Additionally, to handle imperfect training data, a variant of the previous process has been developed which is called soft margins. A box constraint $0 \leq \alpha_i \leq C$ is introduced, and a slack variable ϵ_i is added to the original condition. Therefore, the optimization task transforms to minimizing

$$\frac{1}{2}\mathbf{w} \cdot \mathbf{w} + C\sum_{i=1}^{m} \varepsilon_i$$

subject to

$$y_i(\mathbf{w} \cdot \mathbf{x}_i + b) \geq 1 - \varepsilon_i$$

The modified optimization statement allows, but penalizes, incorrect classification. It requires that both $\sum_{i=1}^{m} \varepsilon_i$ and $\|\mathbf{w}\|^2$ to be minimized. If $\varepsilon_i > 0$ then the object function becomes:

$$L(\mathbf{w}, b, \alpha, \varepsilon) = \frac{1}{2}\mathbf{w} \cdot \mathbf{w} + C\sum_{i=1}^{m} \varepsilon_i - \sum_{i=1}^{m} \alpha_i \left(y_i(\boldsymbol{w} \cdot \boldsymbol{x}_i + b) - 1 + \varepsilon_i\right) - \sum_{i=1}^{m} r_i\varepsilon_i$$

where

$$\alpha_i \geq 0$$

$$r_i \geq 0$$

Let u_i denote the object value:

$$u_i = \sum_{j}^{m} \alpha_i y_i K(\mathbf{x_j}, \mathbf{x_i}) - b$$

then

$$r_i = y_i u_i$$

at the optimum, the derivatives with respect to **w**, b, ε now turn into

$$\frac{\partial \mathrm{L}}{\partial \mathbf{w}} = \mathbf{w} - \sum_{\mathrm{i}=1}^{\mathrm{m}} \alpha_{\mathrm{i}} \mathrm{y}_{\mathrm{i}} \mathbf{x}_{\mathrm{i}} = 0$$

$$\frac{\partial \mathrm{L}}{\partial \mathrm{b}} = \sum_{\mathrm{i}=1}^{\mathrm{m}} \alpha_{\mathrm{i}} \mathrm{y}_{\mathrm{i}} = 0$$

$$\frac{\partial \mathrm{L}}{\partial \varepsilon_i} = \mathrm{C} - \alpha_{\mathrm{i}} - r_i = 0$$

patterns with $0 < \alpha_{\mathrm{i}} < \mathrm{C}$ is referred to as non-bound while $\alpha_i = 0$ or $\alpha_{\mathrm{i}} = C$ are at-bound. The KKT conditions are:

$$r_i \varepsilon_i = 0$$

and

$$\alpha_{\mathrm{i}}(\mathrm{y}_{\mathrm{i}}(\mathbf{w} \cdot \boldsymbol{x_i} + \mathrm{b}) - 1 + \varepsilon_i) = 0$$

The dual object function turns into:

maximizing

$$\mathrm{W}(\boldsymbol{\alpha}) = \sum_{\mathrm{i}=1}^{\mathrm{m}} \alpha_{\mathrm{i}} - \frac{1}{2} \sum_{\mathrm{i,j}=1}^{\mathrm{m}} \alpha_{\mathrm{i}} \alpha_{\mathrm{j}} \, \mathrm{y}_{\mathrm{i}} \mathrm{y}_{\mathrm{j}} \mathrm{K}(\mathbf{x}_{\mathrm{i}}, \mathbf{x}_{\mathrm{i}})$$

subject to

$$\mathrm{C} \geq \alpha_i \geq 0, \forall \mathrm{i}$$

$$\sum_{i=1}^{m} \alpha_i y_i = 0$$

the KKT constraints could be written as:

$$\alpha_{\mathrm{i}} = 0 \leftrightarrow \mathrm{y}_{\mathrm{i}} \mathrm{u}_{\mathrm{i}} \geq 1$$

$$0 < \alpha_{\mathrm{i}} < \mathrm{C} \leftrightarrow \mathrm{y}_{\mathrm{i}} \mathrm{u}_{\mathrm{i}} = 1$$

$$\alpha_{\mathrm{i}} = \mathrm{C} \leftrightarrow \mathrm{y}_{\mathrm{i}} \mathrm{u}_{\mathrm{i}} \leq 1$$

the slake variables do not appear in the soft-margin dual objective formulation. The offset b is calculated by

$$b = y_k - \sum_{i=1}^{m} \alpha_i y_i (\mathbf{x}_i \cdot \mathbf{x}_k)$$

on all non-bound data points. For a novel case, its object value u is calculated, and its class label is determined by the sign of u.

Several parameters need to be tuned when constructing a SVM classifier. Choosing a kernel function has been proved to be important since input data need to be correctly mapped into a feature space. Commonly, Gaussian Radial Basis Function kernels are used. The sigma parameter for this kind of kernel also needs to be tuned. The penalty parameter C determines the balance between training error and generalization error. A large C represents a high penalty for misclassified points. It affects the trade-off between complexity and frequency of error.

2.3 Review of previous studies

Previous studies employed the similar workflow to derive distribution statistics from discrete-return lidar data. Large footprint waveform lidar data have been largely applied to estimate forest biophysical characteristics at the plot-level. Small footprint airborne discrete return lidar provides more details and have been used to model the forest structure at finer scales. Discrete-return lidar data with high point density make it possible to simulate continuous z-value distribution within spatial units. Many research have been done to develop algorithms for lidar classification, tree detection, and model construction under various environmental conditions.

Drake et al., (2002) explained the advantages of large footprint waveform lidar and predicted quadratic mean stem diameter, basal area, and above ground biomass using lidar height metrics at both the plot level (2500-5000m^2) and footprint level (500m^2). They found that the

height of median energy (the height where curve area equals to half the total area) was the best single predictor for above ground biomass. The R^2 value of the regression model predicting above ground biomass was 0.73 at the footprint level and 0.93 at the plot level. They also explained that footprint level relationships were weaker than plot level because of large local variance of forest structure at the footprint scale and geolocation errors.

Regression models developed for one study location usually cannot be used in another location with significantly different environmental conditions and tree species. Drake et al. (2003) examined the generality of the relationships between lidar height metrics and forest characteristics. Stem diameter was measured and used to estimate above-ground biomass for sample plots located in two study areas: a tropical moist forest area in Panama and a tropical wet forest area in Costa Rica. The two study locations have different average rainfall amount. Two metrics were derived from lidar waveforms falling within each sample plot: canopy height and height of median energy (HOME). Simple linear equations were developed between plot-level lidar-metrics and ground-based quadratic stem diameter (QMSD), basal area and estimated above-ground biomass (EAGB). ANOVA was employed to examine the differences between regression equations fitted for each site. HOME was found to be strongly correlated with EAGB, QMSD and basal area in both study areas. Regression model linking EAGB and HOME was more divergent than the regression models linking HOME and the other two forest structural parameters. They concluded that the "differences in the lidar-biomass relationships at the two study areas are primarily the result of the different allometric relationships between stem diameter and above-ground biomass" in the two study areas.

Riggins et al. (2009) estimated forest above-ground biomass from lidar-derived percentile heights. Field measurements were taken in 12 ground reference plots. Total above ground

biomass density (kg/m²) was derived for each plot. A 10 x 10 m neighborhood was created for each grid point within a grid covering the entire study area. The grid point spacing was 1 meter. Percentile heights were calculated from lidar points falling in each neighborhood. Then, the computed percentile heights were used as inputs for image segmentation in eCognition. The segmentation process resulted in polygons representing homogeneous forest structure uints. Lidar metrics were averaged over all pixels within each polygon. The image segments were used to build a regression tree in Cubist for estimating forest biomass. The regression tree was tested on a reserved data set and resulted in a R^2 value of 0.72 and a RMSE of 2.77 kg/m^2.

Clark et al. (2011) combined small footprint discrete return lidar data and a hyperspectral imagery to estimate tropical forest biomass at plot level. Tree measurements were taken in experimental plots. Ground-based plot level aboveground biomass was calculated from individual tree measurements. First returns of the lidar data were transformed to a digital canopy model with a spatial resolution of 0.33 meter. The mean height, mean of the 95^{th} percentile height, maximum height, stardard deviation of heights, kurtosis, skewness, the median to maximum height ratio and the percent of the plot occupied by gaps (cells with a pixel value less than 5 meters were classified as gap cells) were calculated for each plot using pixels within that plot. Plot-level hyperspectral metrics included the red-edge vegetation stress index, plant senescence reflectance index, water band index, normalized difference water index and spectral mixture fractions. Ordinary least squares (OLS) and generalized least squares (GLS) regression models were employed to correlate the plot level metrics with field-derived biomass. Single-predictor and two-predictor regression models were formulated using the statistical language R. The R^2 value ranged from 0.87 to 0.91 and RMSE ranged from 35.8 to 43.2 mg/ha for single-predictor GLS models. The two values ranged from 0.89 to 0.92 and 33.2 to 38.7 separately for

two-predictor GLS models. OLS models occupied similar R^2 and RMSE values. Hyperspectral metrics were also included as predictors in regression models for comparing purposes. Single-variable OLS models' R^2 value ranged from 0.36 to 0.49 and RMSE ranged from 70.3 to 72.5 mg/ha. Two-variable OLS modes' R^2 value ranged from 0.57 to 0.68 and RMSE ranged from 64.4 to 71.3 mg/ha. Models using one lidar metric and one hyperspectral metric had a R^2 value of 0.91 and RMSE value ranging from 35.4 to 36.5 mg/ha. They found that lidar metrics were relatively easier to compute and were highly correlated with forest above-ground biomass. The regression models estimating biomass from lidar metrics were statistically efficient.

Aside from estimating biomass at the plot-level, experiments at the individual tree- level have also been conducted. The first step is usually identifying individual trees in the point cloud. Lidar points associated with identified trees are then used to compute Lidar-metrics. Field measurements of the identified trees are used as independent data for model training. The detection of individual trees from Lidar data is associated with the subject of object detection and image processing. Traditionally, aerial or satellite imagery with high spatial resolution were employed to extract individual trees. The two remote sensing products could be combined for better performance.

Hejun Li et al. (2008) measured tree heights from Lidar point cloud acquired by a ALTM 3100 equipment. They firstly classified the lidar points and generated a digital surface model and a digital elevation model with a spatial resolution of 0.1 x 0.1 meter. The digital elevation model was then used to correct digital true color photos of the study area and generated a digital orthophoto map. A total of 37 individual trees were selected as reference trees. Tree locations were identified using the corrected digital photos and height displacement values. Tree heights were calculated from Lidar dataset and were compared with field-measured tree heights. The

missing of tree tops and aliased points were two main sources of gross errors. Deviation between lidar-measured tree heights and field-measured tree heights was considered to be acceptable. The authors argued that tree heights measured in this way could be used for biomass estimation.

Gleason and Im (2012) compared the predicting precision of four modeling techniques: linear mixed-effects (LME) regression, random forest (RF), support vector regression (SVR) and regression tree (Cubist) on estimating forest biomass. Their experiment was conducted in a 1700 ha moderately dense forest at both the individual tree and plot level. Tree measurements were taken in experimental plots, and plot-level biomass was derived using Jenkins models. Biomass estimation was conducted in four schemes: plot-level biomass estimated from lidar-derived metrics as predictors; plot-level biomass estimation by aggregating the biomass estimated for individual trees within plots; individual-tree level estimation; individual tree-level biomass estimation conducted for coniferous and deciduous trees separately. Tree crown delineation was achieved by using a previous developed algorithm: COTH (a synthesis of genetic algorithm optimized object recognition, treetop identification, and hill climbing). Individual tree-level Lidar metrics included the canopy geometric volumes for 50^{th}, 60^{th}, 70^{th}, and 100^{th} percentiles of height per crown, minimum crown height, 70^{th} and 100^{th} percentile heights per crown, crown area and crown diameter. Plot-level Lidar metrics included leaf area index and canopy geometric volume (sum of individual crown geometric volumes within a plot). The result indicated that all models performed significantly better in the second scheme than in other schemes. SVR provided the most accurate estimation in all four schemes.

Biomass estimation accuracy may diminish in areas with complicated terrains. Yang et al. (2011) simulated the impact of surface topography, footprint size and off-nadir pointing on

vegetation lidar waveforms' shape and vegetation height retrieval using an extended Geometric Optical and Radiative Transfer (GORT) vegetation lidar model. They explained how terrain slope and off-nadir angle distorted canopy height measurements from lidar. Two forest structure datasets and corresponding lidar waveforms were input to the extended GORT model. They concluded that waveform extent stretched as slope and footprint size increased. The increasing of slope made the ground peak less distinguishable from canopy peaks.

One objective of this study is to examine the effectiveness of terrain variation on the accuracy of biomass estimation from lidar data. The slope-induced bias is assumed to be removed by subtracting elevation values from point heights. Thus, extracting ground points from the point cloud is of vital importance. Various lidar classification methods have been proposed and experimented. The classification algorithms vary in terms of suitability (urban area or forest), computing cost, and classification precision. Sithole and Vosselman (2004) reviewed eight filtering algorithms. They found that all filters worked well in landscape of low complexity, and the greatest challenge was to correctly identify complex urban constructions and ground discontinuities.

Jordan et al. (2010) transformed discrete echoes to pseudo-waveforms using cubic spline for a successive of footprints. Lidar metrics were derived from the simulated curves. They argued that the linear regression equations associated with the wave-based metrics are more physically meaningful. They also argued that the traditional frequency-based approach lacks physical explanation on predictor selection. The regression models developed with frequency-based metrics may not be reliable since most of these metrics are highly correlated. The pseudo-wave approach, however, provides explanations on metric selection and captures unique wave patterns which illustrate the forest vertical structure.

3. Methods and data

Lidar data were collected in 2006 and 2011 for Ozark and Trinity River respectively. The lidar point clouds were classified then transformed to height metric layers upon which image segmentation were performed. Ground surveys for the two study sites were conducted in 2006 and 2010 respectively. Above ground biomass derived from *in-situ* measurements and allometric equations were assigned to image segments (objects) as the target variable while lidar-derived height metrics were used as predictors for building regression models.

3.1 Field data

Lidar and ground surveys were conducted at two study sites in different years: Ozark National Forest (ONF), AR and Trinity River (TR), TX. Both study sites were composed of deciduous forests with different dominant tree species and topography. Ground surveys were conducted in 13 plots (Riggins et al., 2009) at the ONF site while 32 plots were sampled at the TR site (Güneralp et al., 2014; Filippi et al., 2014). Diameter at breast height (DBH) and tree species were recorded at both sites. The DBH measurements were input to DBH-based biomass regression equations to estimate plot total biomass.

Jenkins et al. (2003) developed the first national-scale diameter-based biomass regression equations by applying a modified meta-analysis on selected published equations. Previous biomass equations were commonly built with small-size site specific data. Variations in tree component definitions, equation forms and input data requirements also make those equations only applicable to specific sites. There was no formal standard for comparing biomass storage across study sites or regions. The equations proposed in the Jenkins study, on the other hand, are generalizable and consistent. They are applicable to estimate total and component biomass in different regions with different tree species in the United States. "Generalizable" refers to that

the equations are applicable for broad-scale biomass estimation. "Consistent" means that the equations occupy the same tree component definitions, equations forms, and input data requirements (Jenkins et al., 2003).

The first step of the Jenkins study was to search for all available published biomass equations for U.S. tree species. Equations that required tree height or site level measurements other than DBH were excluded. The selected equations used DBH as the single predictor, and total or component biomass as the target variable. The final compilation of biomass equations included 310 total biomass equations and 389 component equations for more than 100 tree species from 104 sources. Tree biomass included in the compilation was from five components: total aboveground, foliage, merchantable stem wood, merchantable stem bark, and coarse roots.

Then, a modified meta-analysis, adopted from Pastor et al. (1984), was used to develop new biomass regression equations from predictions by extracted equations for each tree species group. To be specific, for each regression equation, a certain number of diameter values were selected from all such values originally used to develop that equation. Biomass values were calculated from the diameter values with the equation. This resulted in "pseudodata" for building regression models between aboveground biomass and DBH for each tree species group. The relationships between total aboveground biomass and DBH larger than 2.5 cm were in form of

$$\text{biomass} = e^{\beta_0 + \beta_1 * \ln DBH}$$

Six softwood and four hardwood tree species groups were formed by clustering on taxonomic relationships, wood specific gravity, and diameter-to-aboveground biomass relationships.

At the ONF study site, located in the Ozark National Forest AR (93°55'W 35°42'N, NAD_1983_UTM_Zone_15N), field measurements were taken in 13 plots distributed over the entire study site. Dominant tree species of the sampled area were northern red oak, white oak, maple and hickory. Plot size varies from 624 m^2 to 947 m^2 with irregular shapes. The average plot area is 780 m^2.

Field data were collected in September and October 2006. All trees with diameter at breast height (DBH) larger than 2.5cm were counted within each plot and were classified into four species groups: mixed hardwoods, hard maple/oak/hickory/beech, soft maple/birch, cedar/larch. DBH-based biomass regression equations for the four categories proposed by Jenkins et al. (2003) were employed to calculate aboveground biomass for each sampled tree and summed for plot total biomass. Plot total above ground biomass varies from 8368 kg to 18040 kg, biomass density varies from 10.62 kg/m^2 to 24.61 kg/m^2. Summary statistics for plot-level attributes are listed in the Summary statistics of plot attributes table (**Table 2**).

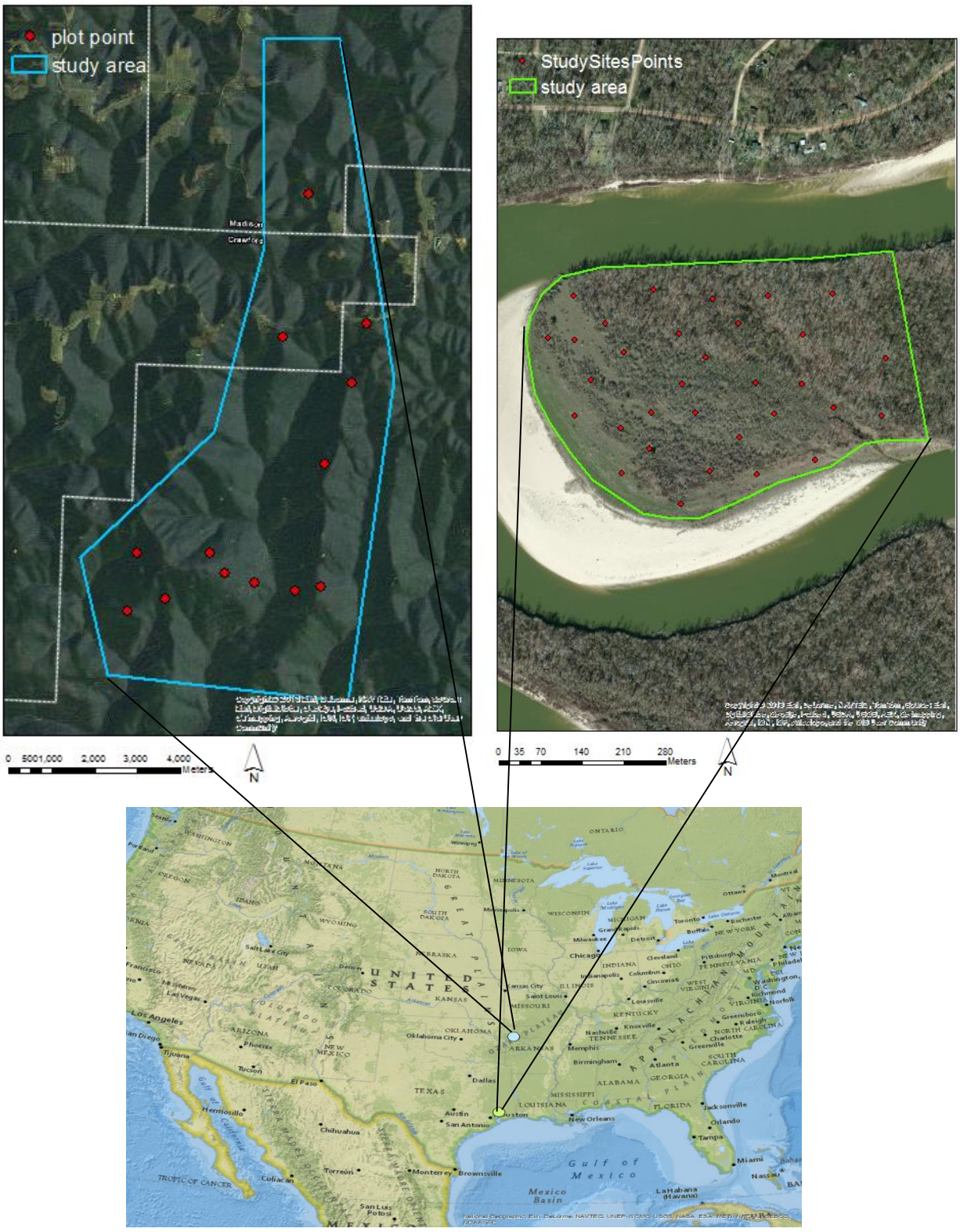


Figure 1 Study site locations

Table 2 Summary statistics of plot attributes (left: Ozark National Forest, right: Trinity River)

Number of Plots	Number of Plots
13	32
Total Biomass(kg)	Total Biomass(kg)
Min:8368 Max:18040 Mean:13590 Std:3035	Min:0 Max:17860 Mean:4387 Std:5102
Biomass Density(kg/m²)	Biomass Density(kg/m²)
Min:10.62 Max:24.61 Mean:17.82 Std:5.19	Min:0 Max:44.75 Mean:10.99 Std:12.78
Area(m²)	Area(m²)
Min:624 Max:947 Mean:780 Std:97	Min:399 Max: 399 Mean:399 Std:0
Ratio of Vegetation Return	Ratio of Vegetation Return
Min:0.52 Max:0.88 Mean:0.78 Std:0.11	Min:0 Max:0.60 Mean:0.34 Std:0.16
Total Number of Lidar Returns	Total Number of Lidar Returns
Min:523 Max:1246 Mean:922 Std:225	Min:461 Max:5237 Mean:1711 Std:1141
Lidar Point Density(points/m²)	Lidar Point Density(points/m²)
Min:0.66 Max:1.49 Mean:1.18 Std:0.25	Min:1.16 Max:13.35 Mean:4.29 Std:2.86

A digital elevation model (DEM) was derived from the lidar data introduced in the next section using TIN interpolation. The surface model is in IMG format with a cell size of 1 meter (**Figure** 3). According to the DEM, each plot occupied significant elevation difference. The terrain of the study site was complex with steep slopes.

The TR study site is a meander-bend at the actively migrating Lower Trinity River TX (94°49'W 30°8'N, NAD_1983_UTM_Zone_15N). The site was composed of mixed deciduous woods with mixed tree ages. Dominant tree species included Chinese tallow, American sycamore, Hackberry and Eastern cottonwood. Ground survey was conducted between mid June and late September 2010 in 32 circular plots with uniform area (399m²). Plots were located on a coordinate grid with variable mesh spacing of 25, 50, and 100m. All trees with DBH larger than 5cm were counted within each plot. Individual-tree total aboveground biomass were calculated from DBH measurements and summed over each plot using both specific allometric equations and biomass regression equations proposed in Jenkins et al. (2003, 2004). Plot total biomass varies from 0kg to 17860kg, biomass density varies from 0kg/m² to 44.75kg/m². **Table 1** lists all summary statistics of plot-level attributes.

A 1 x 1 m digital elevation model (DEM) was also created using TIN method and the lidar data introduced in the next section (**Figure 4**). Analysis on this DEM showed that this site was dominated with typical floodplain terrain. Elevation differences were insignificant.

3.2 Lidar data

Lidar data for the ONF site covering a total area of 32 km² were collected on September 19-20 2006 by 3001 Inc. (Fairfax, VA) using a Leica Geosystems ALS50 sensor. The data were delivered in 36 LAS files, each corresponding to a flight line. The total point number is 71,051,853 with z-value ranging from 218.84m to 1576.58m. The nominal point density is 1.3

points/m². A maximum of 4 returns were collected. Lidar data summary statistics are listed in **Table 3**.

Lidar data for the TR site covered a total area of 27km². The data were collected in June 2011 and were delivered in 12 LAS files with a uniform spatial coverage of 2.25km². The total point number was 96,790,130 with z-value ranging from -3.61m to 209.25m. The nominal point density was 3.6 points/m². A maximum of 4 returns were collected. Lidar data summary statistics are listed in **Table 3**.

Lidar data classification is critical since percentile heights and crown coverage need to be calculated with only high vegetation and ground points. Poor classification blurs the relationship between height metrics and total biomass resulted in biased training sample and weak regression models. Lidar data classification results are shown in **Table 4**. Aside from the DEMs, the lidar data also revealed significant difference between forest structures of the two study sites.

Table 3 Raw lidar Data Summary Statistics

	Ozark National Forest	Trinity River
First Return	51,918,316	63,369,479
Second Return	16,734,271	28,478,087
Third Return	2,399,266	4,719,767
Fourth Return	50,813,027	63,595,052
All	71,051,853	96,790,130

Table 4 Lidar classification results

Ozark National Forest	(code) Classification	#Point (%)	Z Min (meter)	Z Max (meter)
	(1) Unassigned	3,955,268 (5.57)	239.39	744.84
	(2) Ground	12,365,145 (17.4)	239.21	742.74
	(5) High Vegetation	54,685,759 (76.97)	241.83	753.10
	(6) Building	480 (<<0.01)	723.46	730.27
	(7) Noise	45,201 (0.06)	218.84	1576.58
Trinity River	(code) Classification	#Point (%)	Z Min (meter)	Z Max (meter)
	(1) Unassigned	5,099,407 (5.27)	2.43	35.45
	(2) Ground	50,849,310 (52.54)	3.01	17.92
	(5) High Vegetation	40,781,643 (42.13)	5.11	205.64
	(7) Noise	59,770 (0.06)	-3.61	42.14

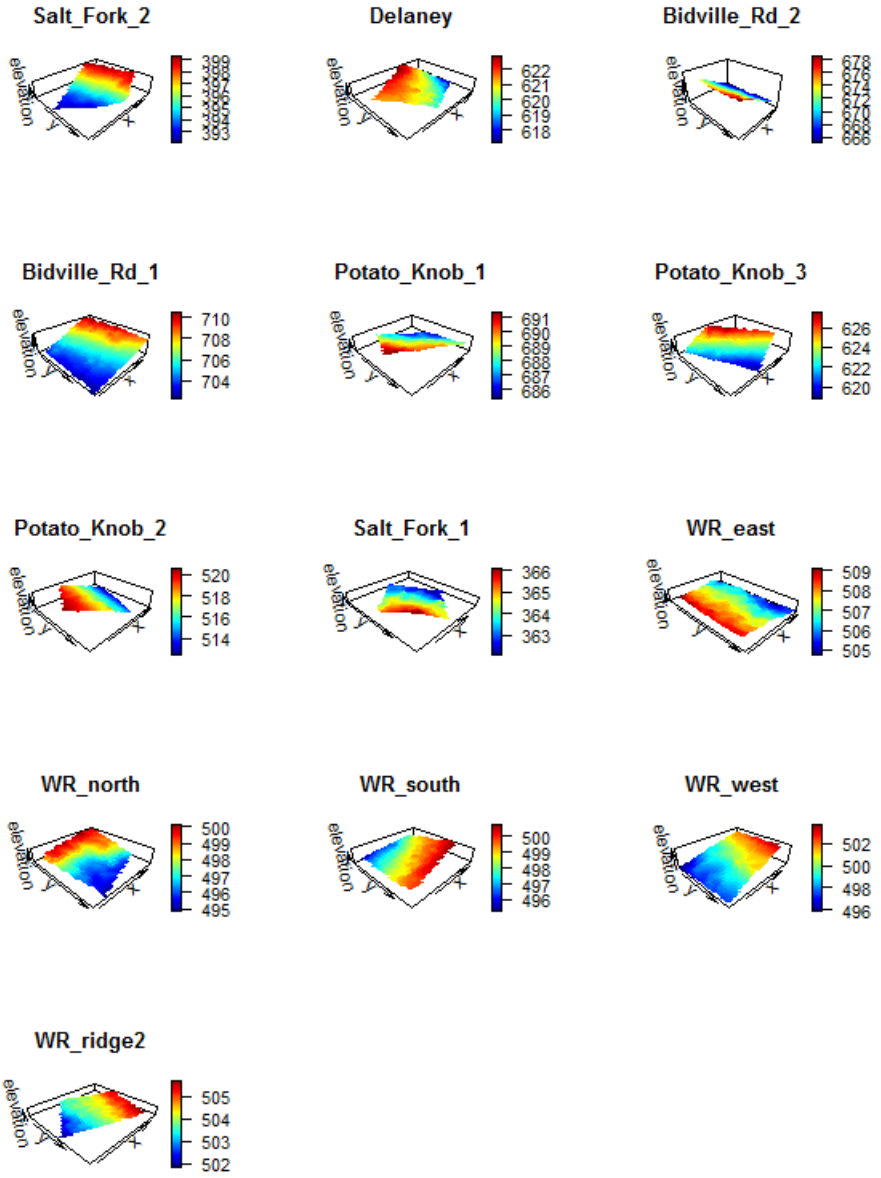


Figure 2 Terrain of each plot at the ONF study site. Elevation and coordinates are in meters. Elevations were extracted from a 1-m DEM created using lidar data. Plot elevation range varies from 4.05m (WR_ridge2) to 14.11m (Bidville_Rd_2) with a mean value of 7.17m and standard deviation of 2.72m.

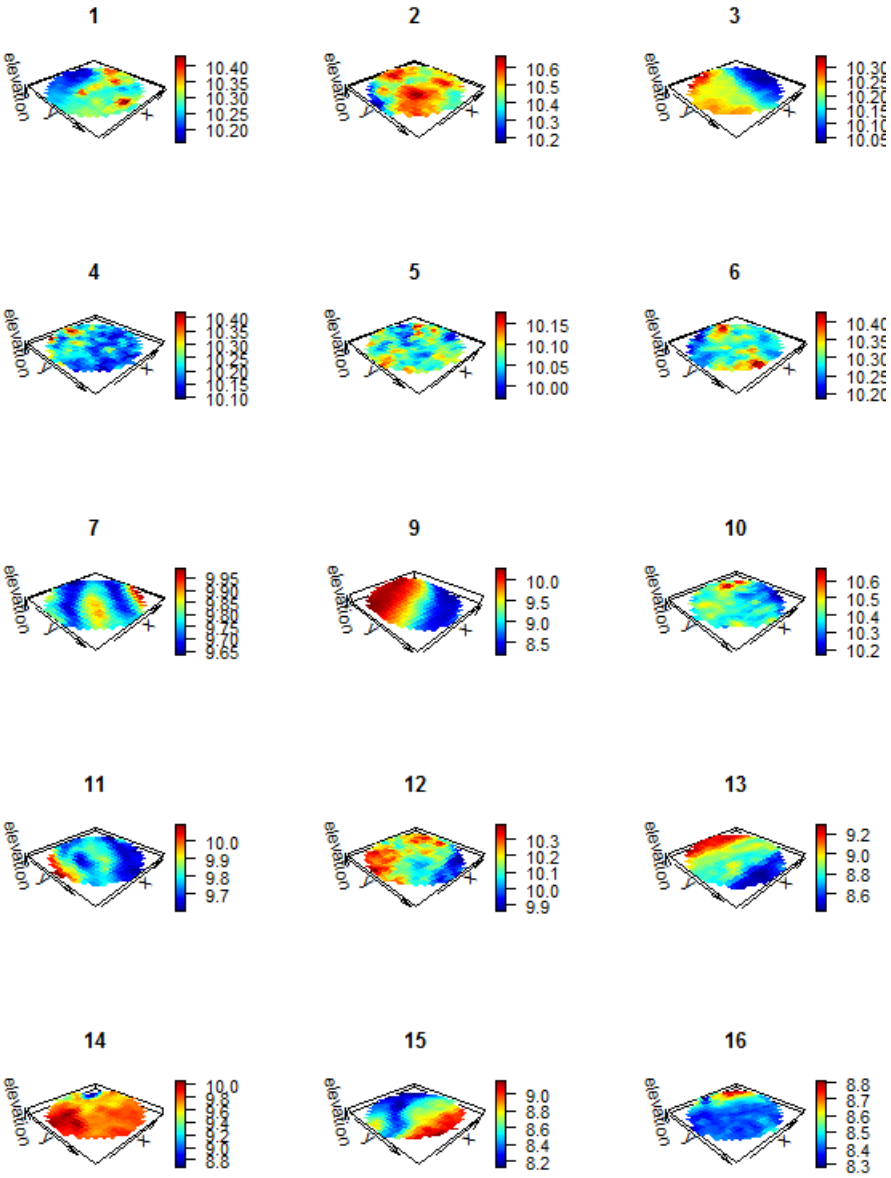


Figure 3 Terrain of each plot at the TR study site. Elevation and coordinates are in meters. Elevations were extracted from a 1 m DEM created using lidar data. The plots were named using their IDs. Plot elevation range varies from 0.31m ("1") to 2.51m ("21") with a mean value of 0.88m and standard deviation of 0.62m.

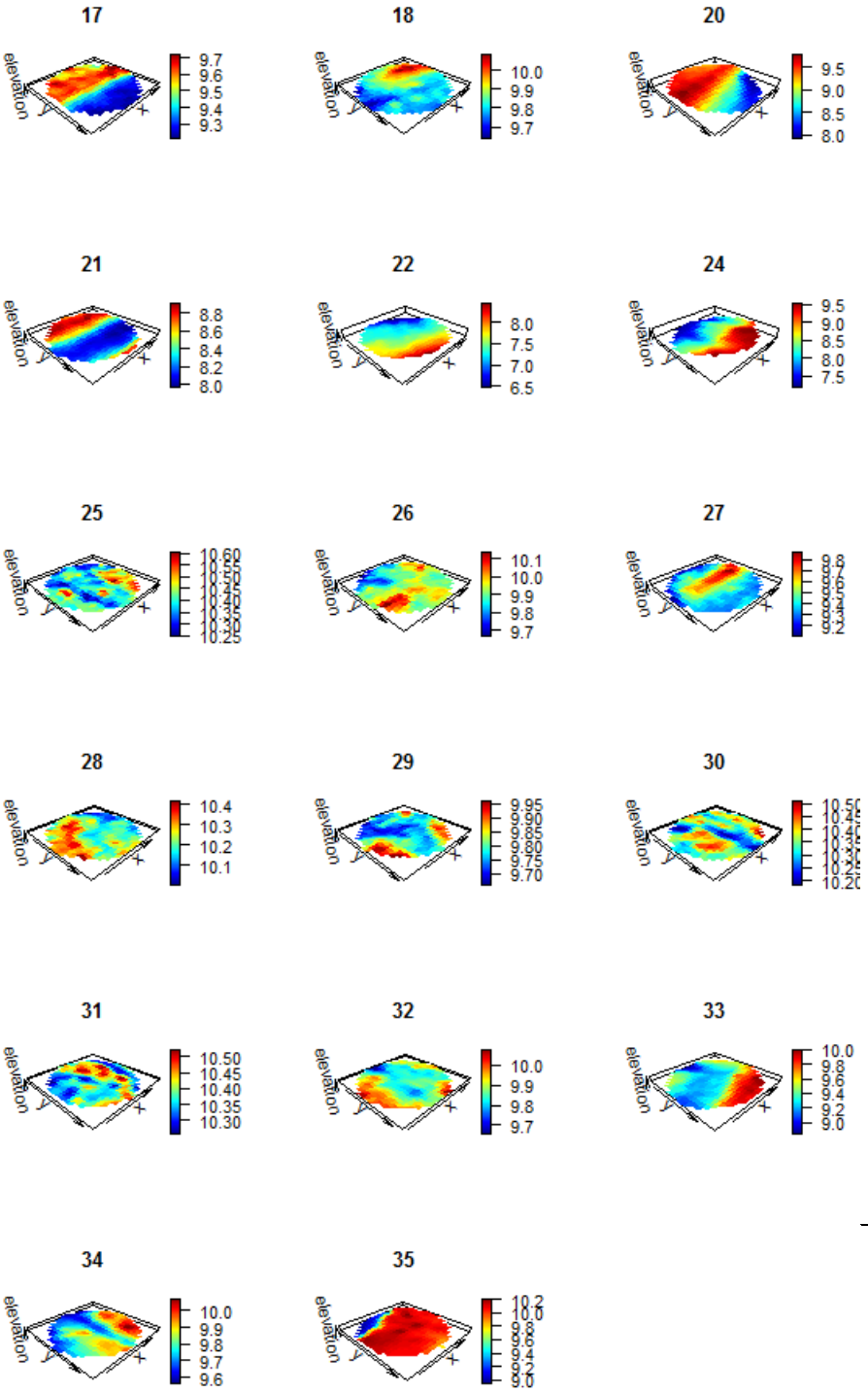


Figure 4 cont. Terrain of each plot at the TR study site. Elevation and coordinates are in meters. Elevations were extracted from a 1 m DEM created using lidar data. The plots were named using their IDs. Plot elevation range varies from 0.31m ("1") to 2.51m ("21") with a mean value of 0.88m and standard deviation of 0.62m.

3.3 Experiment design

3.3.1 Lidar data classification

In this study, percentile heights and canopy ratio were calculated from classified lidar data. Noise points, water points and unclassified points were not taken into account. A robust classification is critical for generating accurate height metrics especially in areas with few lidar points.

Lidar points classified as ground and high vegetation were extracted for deriving height metrics, digital elevation models and canopy height models. Instead of original point heights (the z-values), height above the ground were used for deriving lidar height metrics. The use of relative heights successfully removed the slope-induced bias as illustrated in **Figure 1**.

The lidar processing package LasTools was employed to classify the lidar data. The classification results were bare-eye examined. Lidar data classification results are in **Table 4**.

3.3.2 Calculate lidar height metrics

The neighborhood technique transforms lidar data to 2D images with user-specified resolutions and generalization extents. Image resolution corresponds to the grid spacing parameter of the neighborhood technique which determines the amount of details that could be revealed by the image. The generalization extent is the spatial extent within which lidar points are queried, and is determined by neighborhood size and neighborhood shape.

The neighborhood technique firstly creates a grid covering a specified study extent. Grid points are separated by the pre-defined grid spacing. For each grid point, a neighborhood centering at that point is constructed. Lidar points falling within the neighborhood are extracted

and certain statistics are calculated to describe the distribution of the z-value of these points. The statistics are assigned to the grid point. The grid is then transformed to image layers showing spatial distributions of lidar height metrics.

A large neighborhood leads to reduction in bias introduced by local variance of point distributions. Vegetation structural characteristics would then be less affected by sensor type and lidar survey biases. This characteristic also makes it possible to mimic the return curve of large-scale waveform lidar systems: The lidar points falling within a spatial unit are binned according to their z-values, then the number of points of each bin is used to simulate signal strength (Muss et al., 2011). This procedure results in pseudo-waves which are very similar to wave-form curves in shape and could be processed with methods used for analyzing waveform lidar data.

However, extracting and examining points falling within large neighborhoods requires more computing resource (**Table 5**). A small experiment was conducted to evaluate the computing cost for generating height metrics. A desk-top PC equipped with an 8GB RAM and 3.40GHZ frequency CPU was employed for calculation. Though this experiment was not a rigorous performance assessment, its results clearly showed the increases of time cost as neighborhood size grew.

Additionally, large neighborhoods may mask spatial details making adjacent pixels or grid points very similar to each other. In extreme cases, most lidar points covered by adjacent neighborhoods would be the same leading to severe data redundancy.

Table 5 Compute time for generating height metric layers

Grid Spacing (meters)	Neighborhood Diameter (meters)	Time (seconds) ONF	TR
0.5	5	1359	NA
	10	2062	NA
	15	2892	NA
	20	3950	NA
	30	6706	NA
1	1	252	750
	5	344	1237
	10	515	NA
	15	715	3087
	20	960	4364
	30	1552	7606

Neighborhoods lying close to plots boundaries usually extract lidar points outside the plots. As neighborhoods grow larger, more lidar points outside the plots will be sampled. This may introduce unwanted information to the plots and finally makes regression models unreliable. For example, a plot locates at the boundary between a dense forest and cut woods. Large neighborhoods would possibly bring in structural information of the cut woods. Small plots are especially affected by this phenomenon. When the forest structure is known to be homogeneous around the sampling plots, no biased information would be brought in when using large neighborhoods. The only cost would be higher time and space cost for searching and storing data.

On the other hand, neighborhoods should be larger than grid spacing to allow for overlaps between adjacent neighborhoods. Otherwise, there will be dismissed lidar points leading to possible information lost. Size of overlapping areas is determined by the difference between neighborhood diameter and grid spacing when neighborhood shape is fixed. For example, with prior knowledge that tree crowns are less than 5 meters in diameter and a lidar point density of 10 points per m^2, a neighborhood size of 5 meters will cover sufficient points for

calculation. Fewer lidar points would be extracted when neighborhood size goes smaller. Local details of lidar point distribution would then be more clearly presented leading to possible detections of fine-scale variation of the forest structure under study.

For example, the detection of crown top requires small grid spacing and small neighborhoods. A large neighborhood would blur the height difference between tree crown boundaries and tree gaps surrounding them. The possibility of mixing tree peaks and dimiss of single trees would be increased.

Commonly, a CHM of 1-meter spatial resolution is used as the initial data for detecting crown peaks. Pixel values are assigned to be the largest z-value of all lidar points falling within the pixels. CHMs produced in this way reveals fine spatial details of the crown surface. It is possible that some pixels would lack lidar points and the pixel value would be void. A spatial interpolation process is needed to assign proper values to such pixels.

The other parameter affecting the performance of the neighborhood method is grid spacing. Actually, grid spacing occupies more influence on computing cost and spatial details retained by the neighborhood technique. Large grid spacing results in coarse imagery products while small grid spacing makes it possible to generate images with fine resolutions. More search and extraction process will be performed as the grid spacing goes smaller since more grid points will be used. Additionally, small grid spacing costs more when generating height metric images in addition to more space for data storage.

However, the fine spatial details revealed when using small grid spacing is desirable. When plot size is small, as the case in this study, the increase in computing cost becomes a minor problem. When grid spacing decreases to a certain value, no significant change would be

detected from resultant height metric layers. Thus, a 1-meter grid spacing was chosen to generate height metric imageries in this study.

This study used a fixed grid spacing of one meter and five neighborhood sizes to generate height metric layers from lidar point cloud. Neighborhood shape used in this study was square, and neighborhood size was described in radius which equals to half of the length of a side (**Figure 5**).

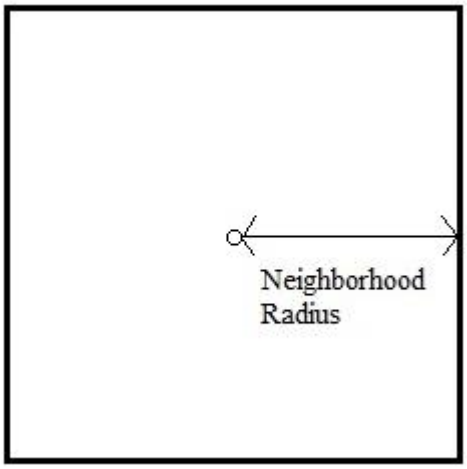


Figure 5 Square neighborhoods

For each plot, a point grid was generated to cover the spatial extent of the plot buffered by two times of the neighborhood size. This approach guaranteed enough lidar points to calculate height metrics for pixels lying close to the plot boundary.

Each parameter set resulted in a multi-band image with 13 bands. The images were clipped by the plot polygons. Therefore, only pixels within the study plots were input to eCogntion for image segmentation. The identified image objects were then compiled as data cases for training and testing regression models. Additionally, a multi-band image of 65 layers was produced by stacking together all the 13-band images. Regression models were also generated using data derived from this image. The image stack was produced with the purpose

of utilizing spatial information captured at various generalization levels. To provide a basis for comparison, all images were clipped by the bounding boxes of plot polygons.

3.3.3 Generate image objects

Image objects were groups of connected pixels and represented homogeneous forest units in this study. The multi-resolution image segmentation algorithm was applied on lidar height metric images. Outputs were polygons each of which corresponded to an image object. The polygons were stored in shapefiles, and were further processed in R. Image segmentation was conducted using eCognition. Input parameters for the multi-resolution image segmentation module are listed in **Table 6**.

Table 6 Image segmentation parameters

Parameter	Value
Use of Hierarchy	1
Starting scale level1	1
Step size level1	1
Starting scale level2	1
Step size level2	3
Starting scale level3	1
Step seize level3	5
Shape	0.5
Compactness	0.5
Number of loops	100

An automatic parameter tuning tool (Estimation of Scale Parameter) was employed to determine the optimal scale parameter (Dragut et al., 2010). Automatic scale parameter detection relies on the observation that an image object matches its real-world counterpart at certain sizes. The ESP tool searches through a sequence of scales and looks for the ones where image heterogeneity stabilizes. It performs image segmentation with each testing scale value and measures image heterogeneity as the mean of local variance of image objects. Image

heterogeneity monotonic increases as scale grows. For certain scales, it stagnates and the first (also the smallest) scale corresponding to a stagnant heterogeneity is taken as the local optimal scale. This study kept the smallest local optimum as larger values led too few image objects in one plot.

The use of hierarchy indicates the algorithm to take into account the object hierarchy which suggests that smaller objects are exactly the components of larger objects. Scale levels indicate that there are multiple real-world objects of increasingly larger sizes that match an image object. ESP searches for local optimums starting from the starting scale levels with steps equal to the step sizes. For each scale within each level, it performs image segmentation on the target image, and calculates the mean local variance over all image objects. The searching process terminates when the number of iteration reaches a user-specified maximum (the number of loops).

Raw outputs were polygon shapefiles with only two fields: shape and id. Other object attributes could be extracted in eCognition when exporting data. This study chose to simply output the two default fields and extract all other attributes using R. Image object polygons were overlaid on the height metric images and the mean of all pixels falling within a polygon was taken as the attribute value for that polygon. Additionally, biomass and biomass density were assigned to every image object based on its vegetation return ratio. This process is referred to as data compilation in this study. Each compiled image object was regarded as a data case for regression algorithms.

Total biomass of an object was derived from the total biomass of the plot it resides in. Plot biomass was considered as the weighed mean of object biomass. Therefore, objects with higher vegetation return ratio were allocated higher biomass. The vegetation return ratio, instead

of absolute value of vegetation return count was used as the weights to eliminate biases induced by spatial variation of lidar sampling density. Biomass density was calculated by dividing total biomass by object area.

3.3.4 Train regression models.

Regression models were trained using three algorithms: SVM, Cubist and Random Forest with R. Each data case occupied 13 predictors and one target variable. Percentile heights below 45 percent were not used, therefore, 9 out of the 13 predictors were actually used for model construction.

To evaluate model performance, a 10-fold cross validation approach was adopted. For each fold, a regression model was trained using data outside this fold. Data inside the fold were used as a validation set. Both training and validation error were reported. An additional model was generated using all available data without partitioning. This approach led to 33 regression models for each neighborhood, 165 models for one study site, and a total of 330 models for the two study sites.

Each model was evaluated by the following statistics: root mean squared error, correlation between predicted target values and observed target values, and adjusted R squared. Each statistics was both calculated on the training set and the validation set. Models generated using all data could only be evaluated using their testing errors. **Table 7, 8** list the statistics calculated for each model.

The R package caret was employed to partition data and tune parameters for the three algorithms. For a model to be tuned, it firstly generates a parameter grid consisting of parameters to be tuned. The input data is also partitioned using a user-specified approach (k-fold

CV, LOOCV, repeated CV, etc.). Caret then applies each parameter combination to train models using a certain cross validation technique. Parameters corresponding to the best model are further used to train models on the entire input data.

3.3.5 Workflow

This study aims to estimate forest above ground biomass at two study sites. The original lidar point cloud were delivered in LAS files. They were transformed to images to extract forest structural information which was further applied to estimate total biomass. Lidar-derived percentile heights and canopy coverage have been proved to be robust predictors for biomass, thus, they were used as predicting variables in this study. Horizontal and vertical coordinates and classification information of the lidar points were utilized to calculate percentile heights and canopy coverage as mentioned in previous sections.

The neighborhood technique was the core for converting discrete points to multi-band images. The neighborhood size determined the number of lidar points that could be sampled at each grid point. Therefore, it significantly influenced the value of height metrics. Briefly speaking, too large a neighborhood blurs spatial details while a two small one extracts very limited lidar points for calculation. The other parameter, the grid spacing, is much easier to determine. Smaller grid spacing corresponds to finer images. The balance between fine resolution and computing cost was made and a one meter resolution was adopted in this study.

There's no previous study indicating any rigorous methods or equations for determining grid spacing and neighborhood size. Thus, this study derived regression models on dataset generated using 5 neighborhood sizes ranging from 0.5 to 15 m as shown in **Figure 6**. The model performances were used to evaluate the appropriateness of the neighborhood sizes. Input parameters for the multi-resolution image segmentation algorithm, the neighborhood technique

and machine learning techniques need to be tuned for better performance of regression models. Thus, this study employed a parameter grid to examine the effects of different parameter combinations on model performance.

Summarily, this study generated height metric images using different neighborhood sizes and built regression models using data converted from the images. The effects of both parameters for the neighborhood technique and regression algorithms were evaluated according to model performances.

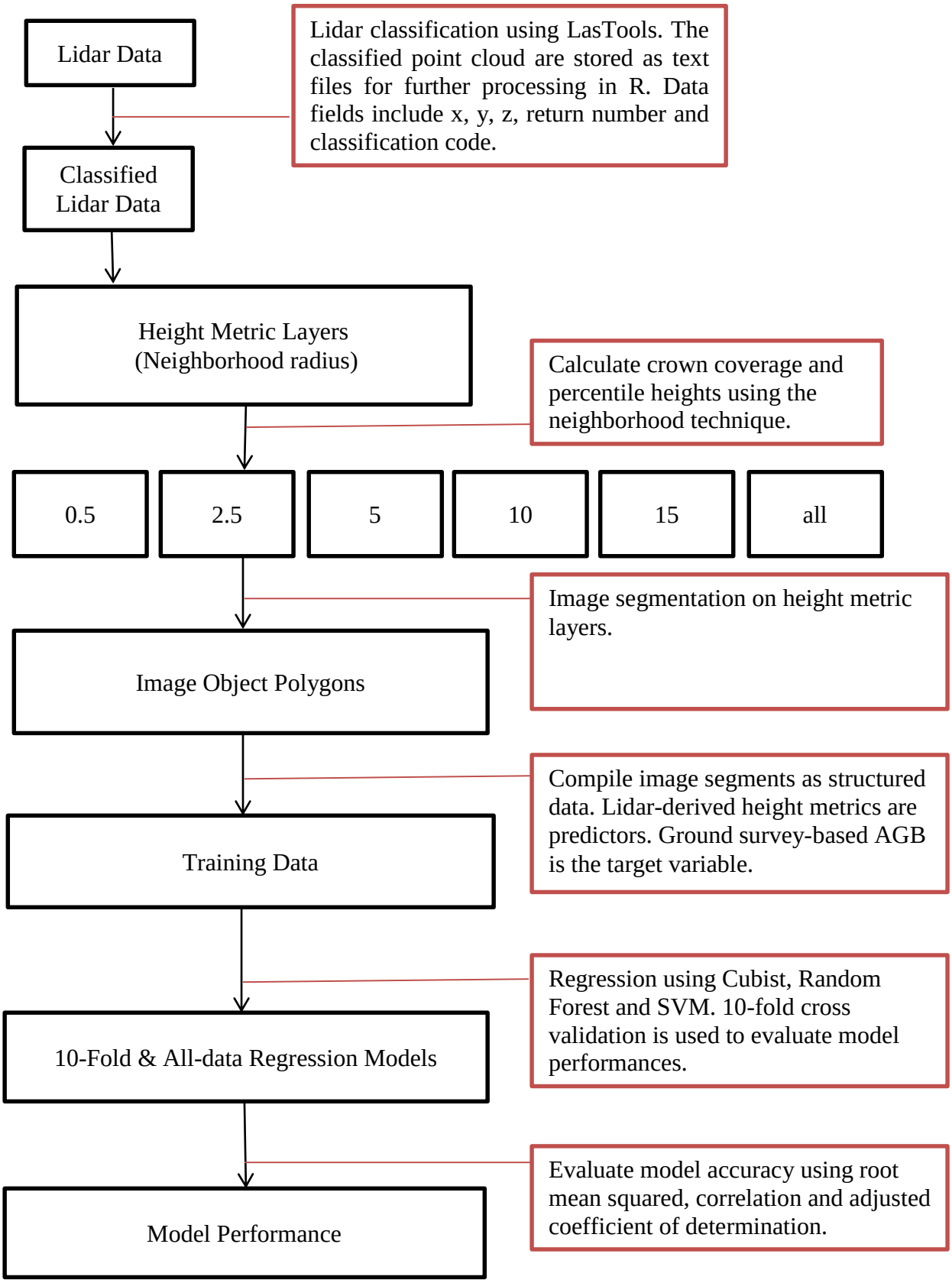


Figure 6 Workflow

3.4 Software

LAS files were classified and transformed to text files using LASTools. The text files were read into statistical software R as data frames. Percentile heights and corresponding height metric images were created in R and written to the file system. Image segmentation was conducted in eCognition, and Dr. Lucian's ESP tool (Dragut et.al. 2010) was employed to automatically select the scale parameter. Outputs of the image segmentation procedure were image objects stored as polygons in shapefiles. These image objects were compiled as structured cases in R. Regression models were trained and tested in R. Statistics for evaluating model performance were also calculated in R. The R package caret streamlines model training and testing. It relies on a number of R packages to build models and tune parameters for these models using some cross validation techniques.

In this study, the package Cubist was used by caret to build Cubist regression trees. The packages was written by Max Kuhn, Steve Weston, Chris Keefer and Nathan Coulter and maintained by Max Kuhn. The function "cubist.default" offered by the package implements the GPL version of the C code given on the RuleQuest website (Max Kuhn, 2014). Package "randomForest" written by Leo Breiman, Adele Cutler, Andy Liaw and Matthew Wiener and maintained by Andy Liaw was invoked in caret to implement the Random Forest algorithm. Support vector regression was implemented by package e1071. The package was written by David Meyer, Evgenia Dimitriadou, Kurt Hornik, Andreas Weingessel and Friedrich Leisch and maintained by David Meyer. Functions embedded in e1071 were indirectly utilized through package kernlab in caret. kernlab was written by Alexandros Karatzoglou, Alex Smola and Kurt Hornik and is maintained by Alexandros Karatzoglou.

4. Results and Discussions

Lidar height metric images generated with different neighborhood sizes showed remarkable variation. The height metric images were segmented using the multi-resolution algorithm. The output was six sets of image objects (stored as shapefiles) whose attributes were extracted in R. This study assessed sensitivity of model performance to the neighborhood size. Model performances (**Figure 7,8,910**) were measured by three statistics on both training and validation set: root mean squared error (RMSE), correlation between predicted and actual values (COR), and adjusted R^2 (RSQ-adj). Two approaches were adopted to build regression models using each of the three algorithms (SVM, Cubist, RandomForest). The first approach built regression models using 10-fold cross validation and generated 10 models for each algorithm. Each of these models was evaluated on both training (training RMSE, training COR, training RSQ-adj) and validation data (validation RMSE, validation COR, validation RSQ-adj). The second approach uses all data to build regression models which were evaluated by RMSE, COR, and RSQ-adj). Using cross validation reduces the risk of over-fitting and reveals a more restricted estimation of model performance on unknown data. The experiment results shown in A1 and A2 demonstrate that model performance can be unreliably estimated by only one data partition.

The underlying assumption is that as neighborhood size grows from very small to large, regression models become increasingly robust up to a stable level. Then, as the neighborhood size continues to grow, model performance decreases. When the neighborhood size is smaller than a certain threshold, height metrics are biased by lidar survey inaccuracies. The lack of lidar points not only leads to lack of points for calculating height metrics but also introduces unwanted minor details. A too large neighborhood, on the other hand, not only requires much more

computational resource but also masks spatial details and makes height metrics insensitive to biomass variations. According to the assumption, unfeasible neighborhood sizes leads to biased relationships between biomass and height metrics. Regression models will likely to give accurate predictions on the training data but generalize badly on the validation data. Thus, algorithms may seem to over-learn while the poor validation accuracy may actually stem from the low quality of training samples. Therefore, a good neighborhood size should correspond to models with high and similar training and validation COR and RSQ-adj. A sensitive and robust algorithm should be able to learn from the training data and also be able to select the most appropriate neighborhoods.

4.1 Experiment results at the ONF study site

For the 0.5 meter neighborhood radius, SVM and Random Forest models occupy high training COR and RSQ-adj. Cubist models made unreliable predictions on the 10-fold training data (an average RSQ-adj of 0.158). All three algorithms generalized badly on the validation data. The performance of SVM and Cubist models on individual folds differ a lot while SVM models produced more homogeneous evaluation statistics (coefficient of variation equals to 0.47 and 0.90 respectively). Random Forest is robust on the training data but it also produces the least accurate model on the validation data. Additionally, there's no significant correlation between predicted values and observed values at this neighborhood size over all three algorithms on the validation data (average validation RSQ-adj are below 0.1). Cubist models performed much better on the 10-fold training data with a neighborhood size of 2.5 meters (average training RSQ-adj increases to 0.671). SVM produced significantly lower training accuracy (average training adjusted R^2 equals to 0.598). Random Forest produced almost the same average training RSQ-adj as it did on the 0.5-meter neighborhood and its validation accuracy increases

significantly (average training RSQ-adj equals to 0.895, validation RSQ-adj equals to 0.419). This result indicates that the 2.5-meter neighborhood captures more forest structural information than the 0.5-meter neighborhood but it is still too small to get robust biomass predictions as the average validation COR and RSQ-adj of the three algorithms are low.

When the neighborhood radius grows to 5 meters, a sudden drop of model performance on the training data is observed for SVM while model evaluation statistics for Cubist and Random Forest only change slightly. The drop makes training and validation accuracy very close for SVM indicating that the SVM models successfully revealed true relationships between height metrics and total biomass, while models built using the other two algorithms are biased and unreliable since their validation accuracies are significantly lower than the training accuracies (**Figure 8**). However, both validation COR and RSQ-adj are low for all three algorithms indicating the five-meter radius is not an optimal choice (validation RSQ-adj of the three algorithms equal to 0.397, 0.273, 0.350 respectively).

For the 10-meter neighborhood data, all three algorithms produced impressively high predicting accuracy on both training and validation data. SVM models generated high average training COR (0.980) and RSQ-adj (0.959). The validation COR ranges from 0.569 to 0.972 with an average of 0.865. Validation RSQ-adj ranges from 0.257 to 0.982 with an average of 0.738. The performance of Random Forest on the training data is similar to SVM but its validation accuracy is much lower (average training RSQ-adj equals to 0.928, average validation RSQ-adj equals to 0.454).

Cubist models also performed well on the training data. The models occupy high training accuracy with training COR ranging from 0.912 to 0.962 and training RSQ-adj ranging from 0.830 to 0.925. The validation accuracies are significantly lower while the average validation

RSQ-adj still reaches 0.536 which is much higher than previous neighborhood settings. Therefore, the Cubist models over-learned the training data but still be able to predict biomass values on the validation data with acceptable accuracies.

The largest neighborhood radius tested in this study is 15 m. All three algorithms produced robust predicting results at this scale. All the models occupy very high training and validation COR and RSQ-adj. SVM models show the best performances with an average validation COR of 0.957 and average validation RSQ-adj of 0.902. Cubist models and Random Forest also produced high accuracy with average training RSQ-adj larger than 0.95 and average validation RSQ-adj larger than 0.75. Additionally, training and validation accuracy of SVM is close to each other (**Figure 8**).

For the image stack containing all image layers generated using the five neighborhood sizes, Random Forest occupies the highest average training COR and RSQ-adj (0.964, 0.929), while Cubist gets the lowest (0.662, 0.483). SVM shows the highest average validation COR and RSQ-adj (0.675, 0.474) while Random Forest has the lowest (0.500, 0.253). Height metrics generated under infeasible neighborhood settings introduced unreliable relationship to models fitted using the image stack, and they should not be included in the multi-dimensional analysis.

For each neighborhood setting, all data cases were also used to train a regression model without data partitioning, and thus, without validation model evaluation statistics. The estimated predicting accuracy is significantly biased from either single-fold training or validation accuracy or the average values. This is obvious at the 0.5-meter neighborhood where the COR and RSQ-adj of SVM on all data cases reach 0.998 and 0.995 respectively while the validation COR and RSQ-adj are only 0.182 and -0.005. However, the three set of models performed significantly worse on the 10-fold validation data.

The 15 m neighborhood and SVM are considered to be the most feasible combination for the Ozark National Forest study site regarding the estimation accuracies of all models considered (**Figure 8)**. Estimation residuals (observed minus predicted) for the SVM model built on all data with 15 m neighborhood were examined (**Figure 11**). The figure was plotted on biomass density, and were individually plotted because sampling plots are distributed across the entire study site (**Figure 2**). Distance between two plots varies from 594 to 10630 m with a median of 4153 m. Residual ratio is defined as the absolute of the ratio between residuals and observed values. **Figure 12** shows the distribution of residual ratio over segments (data cases). The median of the residual ratio is 0.08 over the 81 segments indicating that the residuals are small in general. This is consistent with the model performance at the 15 m neighborhood (**Figure 8** and **Appendix 1**). However, there exist extreme values as nine objects whose residual ratio are larger than 0.3. Five out of the nine have residual ratios that are larger than 1. Four out of the five objects occupy no vegetation returns and zero total biomass and biomass density, however, the use of neighborhoods counts for vegetation returns outside the segments which is a problem of using big neighborhoods. The other one object has a residual ratio of 1.01 and an area of 8 m^2. Its total biomass is very low (181 kg) comparing to other objects in the same plot (median total biomass equals to 628) as well as to objects over the entire study area (median total biomass equals to 1986). Objects whose residue ratio are larger than 0.3 reside in plot "WR_ridge_2" and are small in size (object area varies from 8 to 57 m^2 while the median object size is 71 m^2). This indicates that large neighborhoods may introduce significant bias to biomass estimates in small objects. It also suggests that very small objects may need to be merged to larger objects to improve accuracy.

Summarily, the neighborhood size significantly influenced the quality of training samples. Cross validation should be used for a more restricted estimation of model performance. The 15 m neighborhood generated the best models since both training and validation accuracies are high and are close (**Figure 8**). SVM is the best algorithm to build regression models for this data set. Random Forest over-learned on all data sets as the model accuracy table shows.

4.2 Experiment results at the TR study site

For the 0.5-meter neighborhood, single-fold predicting accuracies of SVM models on the testing data do not show much variation (coefficient of variation of COR and RSQ-adj equals to 0.022 and 0.045 respectively). Single fold validation accuracies are lower than the training accuracies and show larger variation (coefficient of variation of COR and RSQ-adj equals to 0.186 and 0.119). The other two algorithms show the same pattern. Average validation accuracies of the three algorithms are similar (average RSQ-adj of SVM, Cubist and Random Forest are 0.537, 0.568, 0.523 respectively), while average training accuracies show great divergence (average RSQ-adj equals to 0.659, 0.820, 0.921 respectively). This indicates that Random Forest mostly over-estimated the data while SVM gave the most restricted accuracy estimation. The low validation accuracies also suggest that 0.5 meter is feasible for estimating biomass at this study site though high training accuracies have been observed.

Regression models built on the 2.5-meter neighborhood also show larger variation on the validation data than the training data while validation evaluation statistics are significantly larger comparing to the ones in the 0.5-meter neighborhood. The average validation RSQ-adj reaches 0.720, 0.784 and 0.758 for SVM, Cubist and Random Forest. Meanwhile, the average training RSQ-adj reaches 0.900, 0.931 and 0.954. As the neighborhood size grows to 5 meters, validation accuracies drop significantly while training accuracies remain high for Cubist and

Random Forest as **Figure 10** shows. For SVM, the average training COR and RSQ-adj drop from 0.949 to 0.846 and from 0.900 to 0.716 indicating that it is sensitive to the change of neighborhood size. When the neighborhood size grows to 10 meters, both validation COR and RSQ-adj grow much larger for all three algorithms (**Figure 10**). For the 15-meter neighborhood, training accuracies of all three algorithms remain significantly higher than the validation accuracies.

Furthermore, Random Forest shows stable training average accuracy over all neighborhood settings while validation accuracy varies significantly ranging from 0.430 (5 meters) to 0.805 (15 meters). Cubist shows the similar pattern as **Figure 10** indicates. When compared to the image stack containing all image layers, the five neighborhood settings generated lower average validation accuracy and larger difference between training and validation accuracy. Single-fold models built on the image stack also show low accuracy variation.

Estimation residuals of the most accurate model (learning algorithm is SVM, data features are percentile heights generated on neighborhoods from 0.5 to 15 m) are shown in **Figure 12**. The graph was plotted on biomass density. Out of the 301 objects, 106 objects' residual ratio are larger than 0.3. There are 56 objects whose residual ratio are larger than 1 most of which occupy very low vegetation returns, and 39 occupy zero vegetation returns. As for ONF, this indicates that large neighborhoods are likely to introduce unwanted information to single objects, and the negative effect of this characteristic becomes more severe as object size decreases.

Summarily, 2.5, 10, 15 m neighborhood data sets and the combined data set offer reliable training samples to build regression models. SVM is most unlikely to over-learn from the data

and gives most reliable model performance estimation. Unlike the ONF study site, combining all image layers at TR generated more accurate regression results on both training and validation data. Therefore, it is feasible to use all image layers to train regression models. Additionally, as SVM generated the highest training and validation accuracy on the combined image layers, it should be selected for biomass estimation at TR.

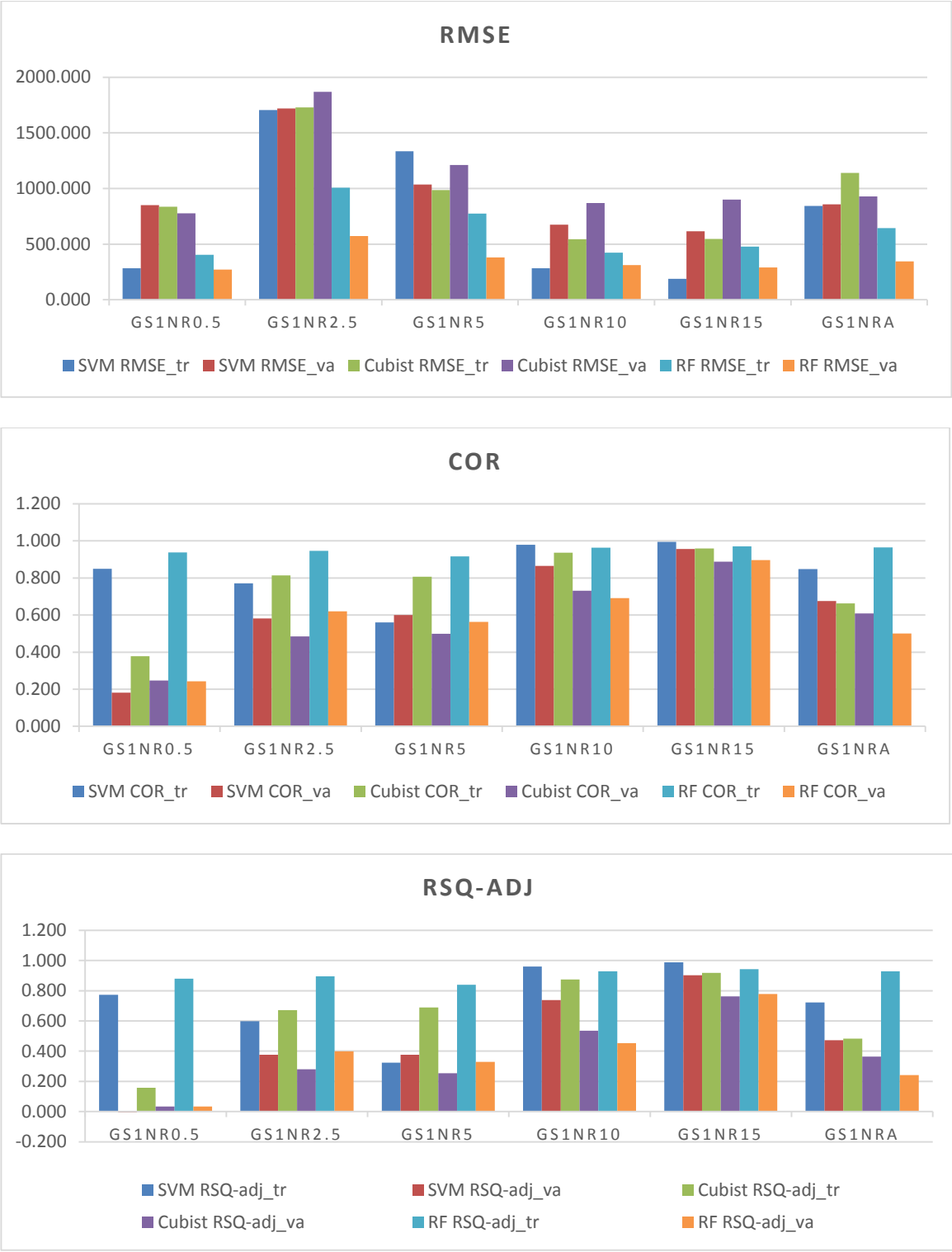
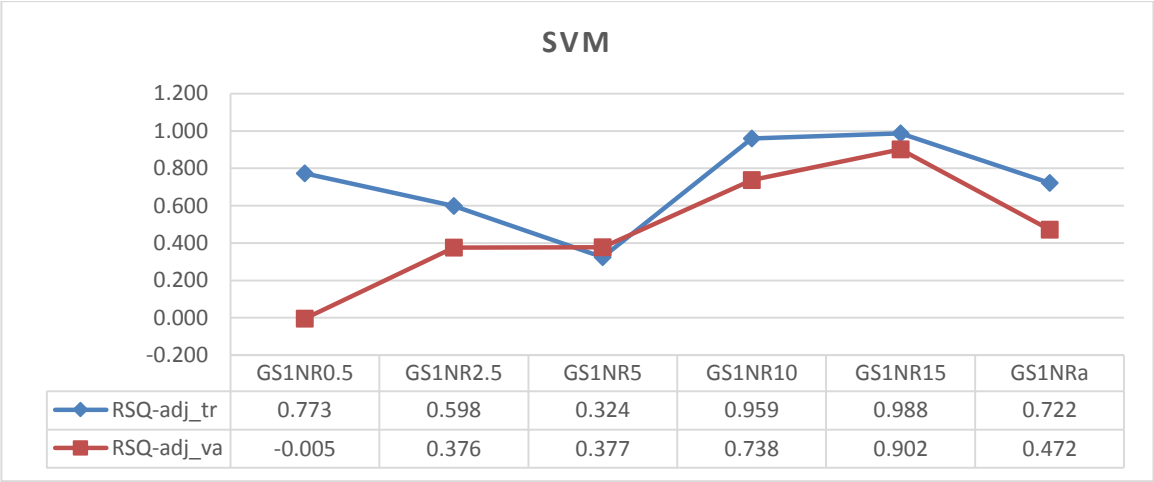
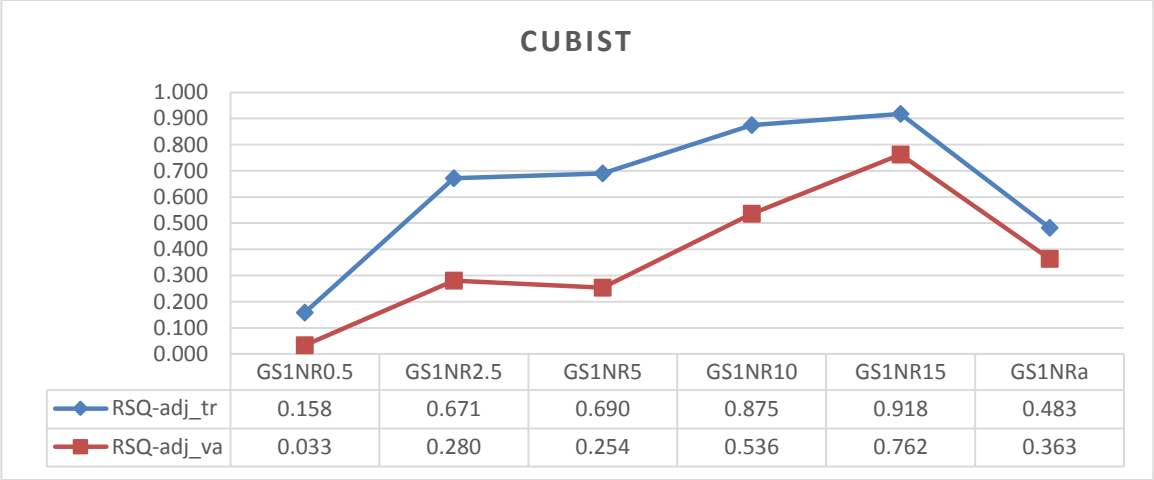


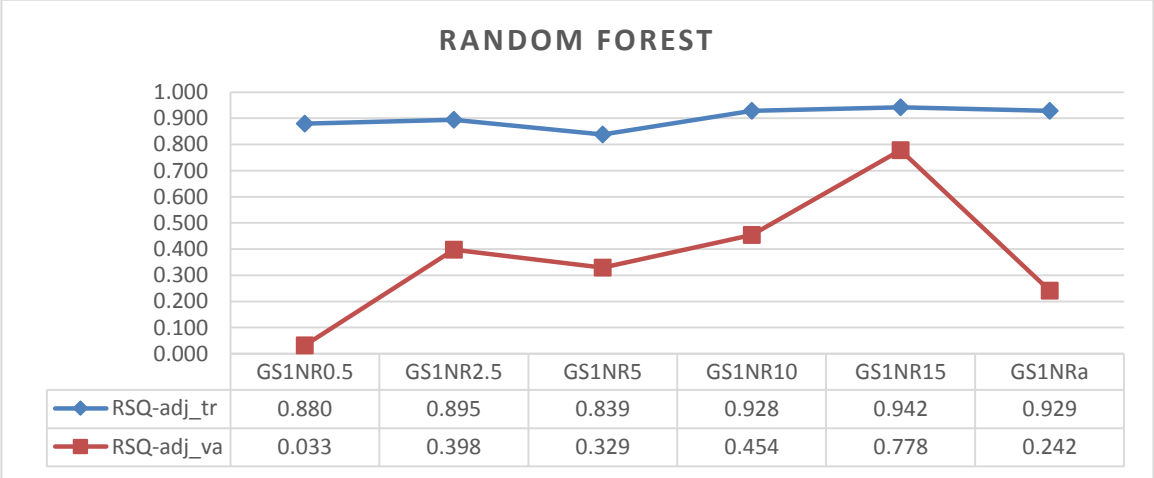
Figure 7 Model evaluation statistics at Ozark National Forest Site. Each graph corresponds to one evaluation statistics. Both training and validation statistics are listed.



	GS1NR0.5	GS1NR2.5	GS1NR5	GS1NR10	GS1NR15	GS1NRa
RSQ-adj_tr	0.773	0.598	0.324	0.959	0.988	0.722
RSQ-adj_va	-0.005	0.376	0.377	0.738	0.902	0.472



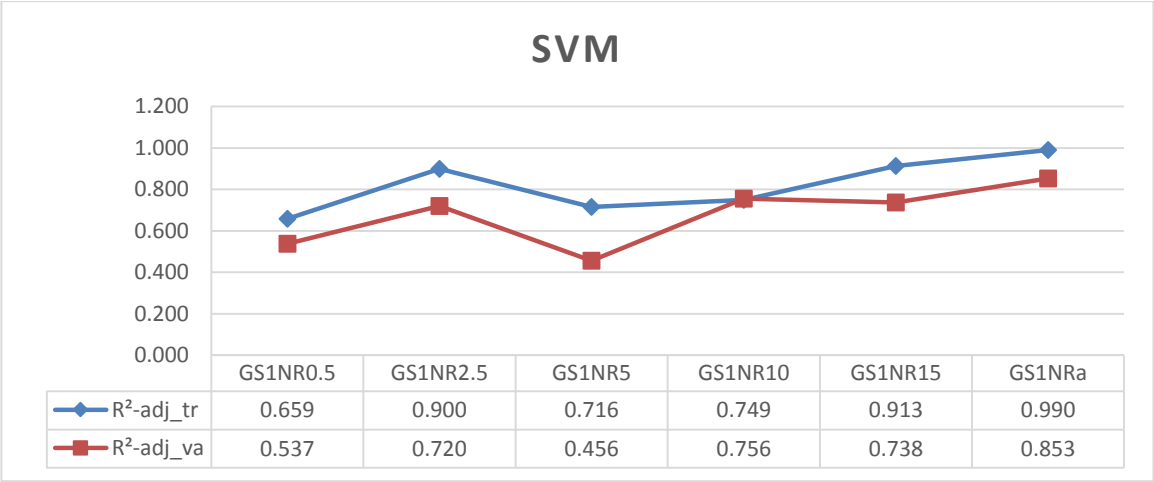
	GS1NR0.5	GS1NR2.5	GS1NR5	GS1NR10	GS1NR15	GS1NRa
RSQ-adj_tr	0.158	0.671	0.690	0.875	0.918	0.483
RSQ-adj_va	0.033	0.280	0.254	0.536	0.762	0.363



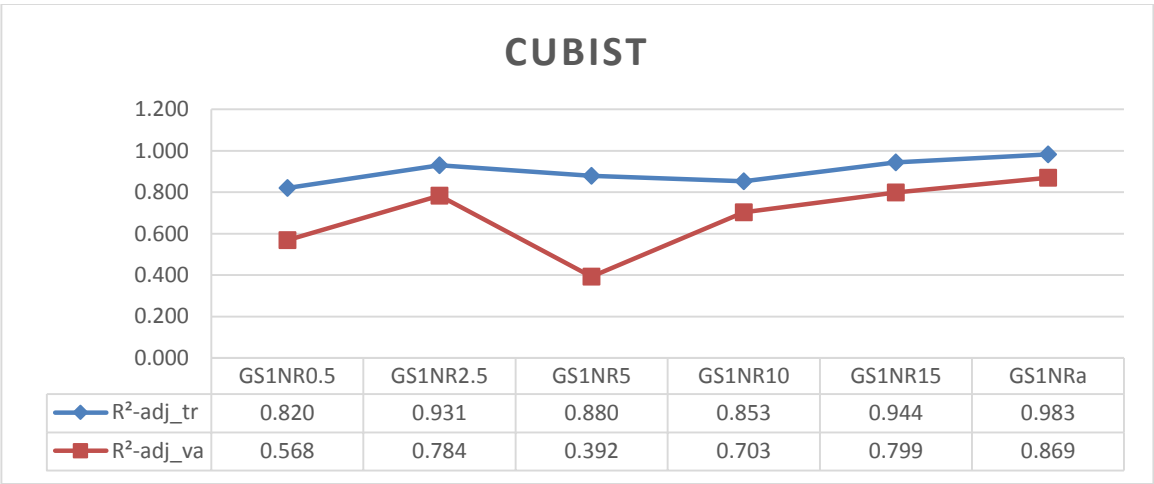
	GS1NR0.5	GS1NR2.5	GS1NR5	GS1NR10	GS1NR15	GS1NRa
RSQ-adj_tr	0.880	0.895	0.839	0.928	0.942	0.929
RSQ-adj_va	0.033	0.398	0.329	0.454	0.778	0.242

Figure 8 RSQ-adj at ONF. Each graph shows the variation of RSQ-adj of one regression technique over different neighorhood sizes. Each node cooresponds to a bar in the third graph in **Figure 7**.

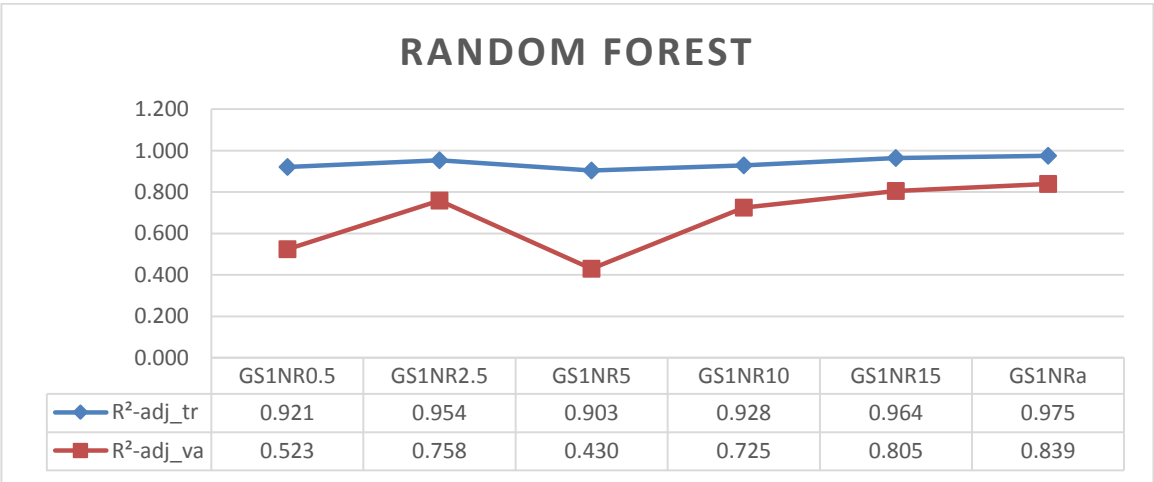
Figure 9 Model evaluation statistics at Trinity River site.



	GS1NR0.5	GS1NR2.5	GS1NR5	GS1NR10	GS1NR15	GS1NRa
R^2-adj_tr	0.659	0.900	0.716	0.749	0.913	0.990
R^2-adj_va	0.537	0.720	0.456	0.756	0.738	0.853



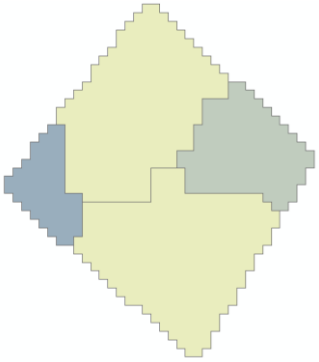
	GS1NR0.5	GS1NR2.5	GS1NR5	GS1NR10	GS1NR15	GS1NRa
R^2-adj_tr	0.820	0.931	0.880	0.853	0.944	0.983
R^2-adj_va	0.568	0.784	0.392	0.703	0.799	0.869



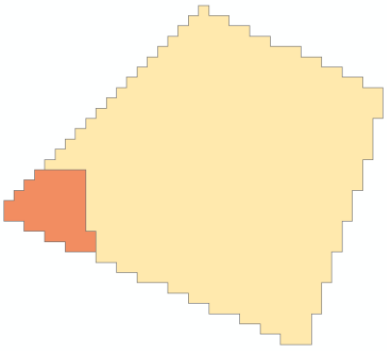
	GS1NR0.5	GS1NR2.5	GS1NR5	GS1NR10	GS1NR15	GS1NRa
R^2-adj_tr	0.921	0.954	0.903	0.928	0.964	0.975
R^2-adj_va	0.523	0.758	0.430	0.725	0.805	0.839

Figure 10 R^2-adj at Trinity River site.

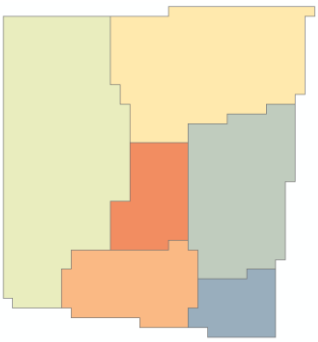
Salt_Fork_1



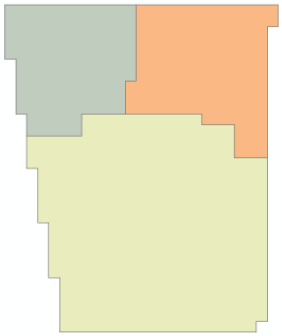
Salt_Fork_2



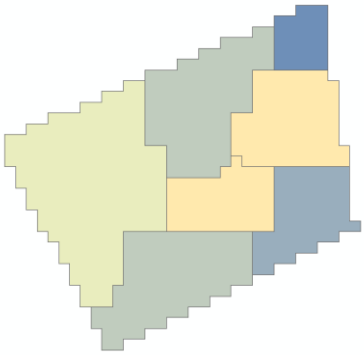
Bidville_Rd_1



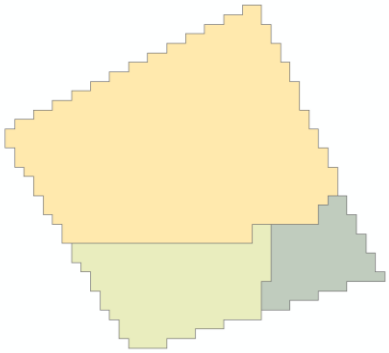
Bidville_Rd_2



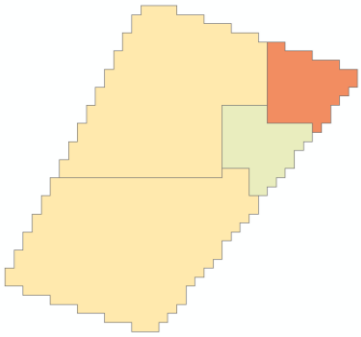
Potato_Knob_1



Potato_Knob_2



Potato_Knob_3



Delaney

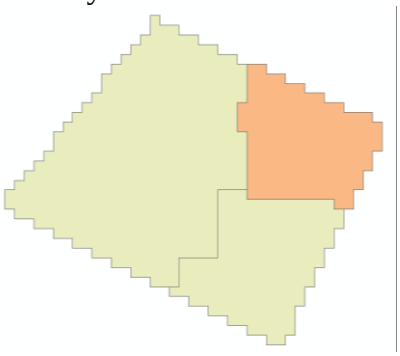


Figure 11 Support vector machine (SVM) biomass regression residuals at Ozark National Forest site. This regression model was built on a 15 m neighborhood without data partitioning.

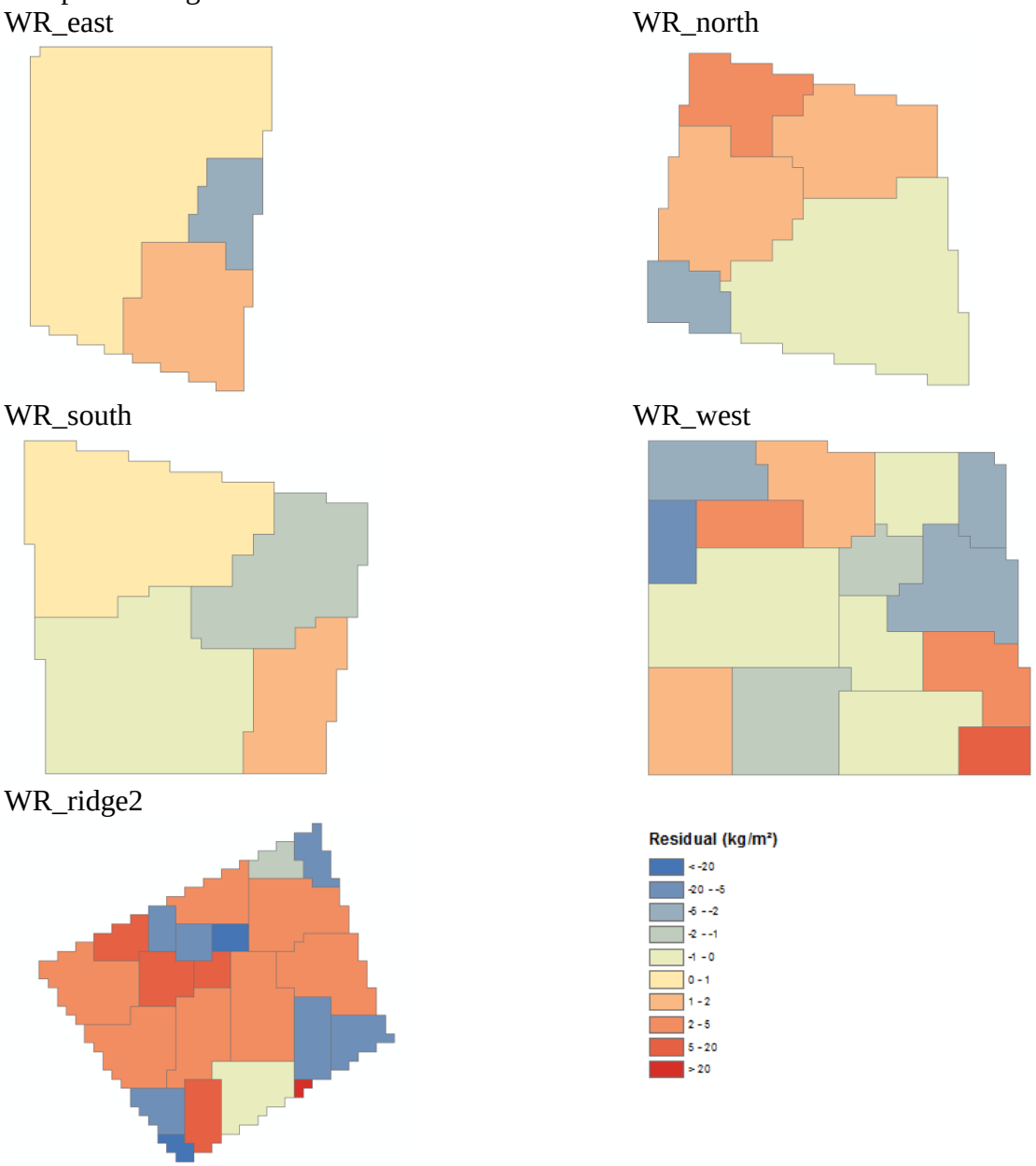


Figure 11 cont. Support vector machine (SVM) biomass regression residuals at Ozark National Forest site. This regression model was built on a 15 m neighborhood without data partitioning.

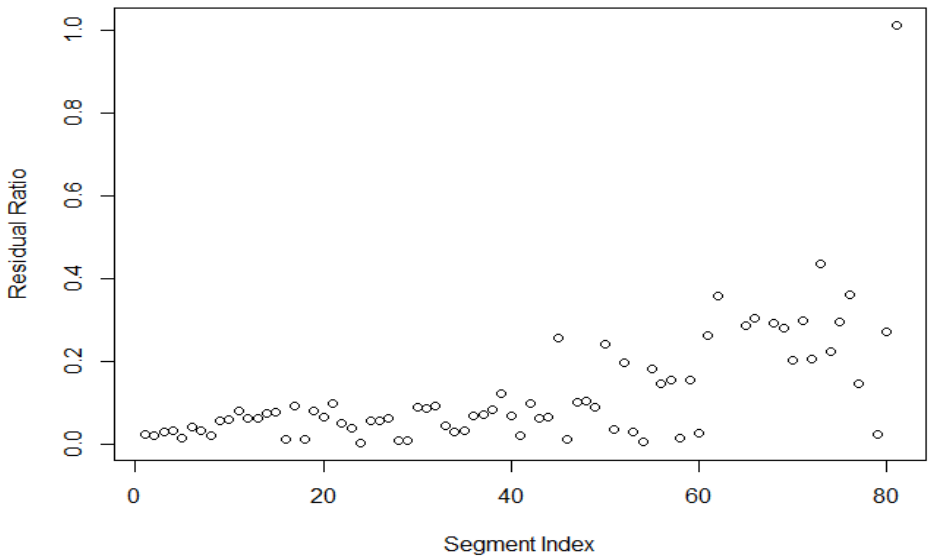


Figure 12 Estimation residuals of the selected model at Ozark National Forest site.

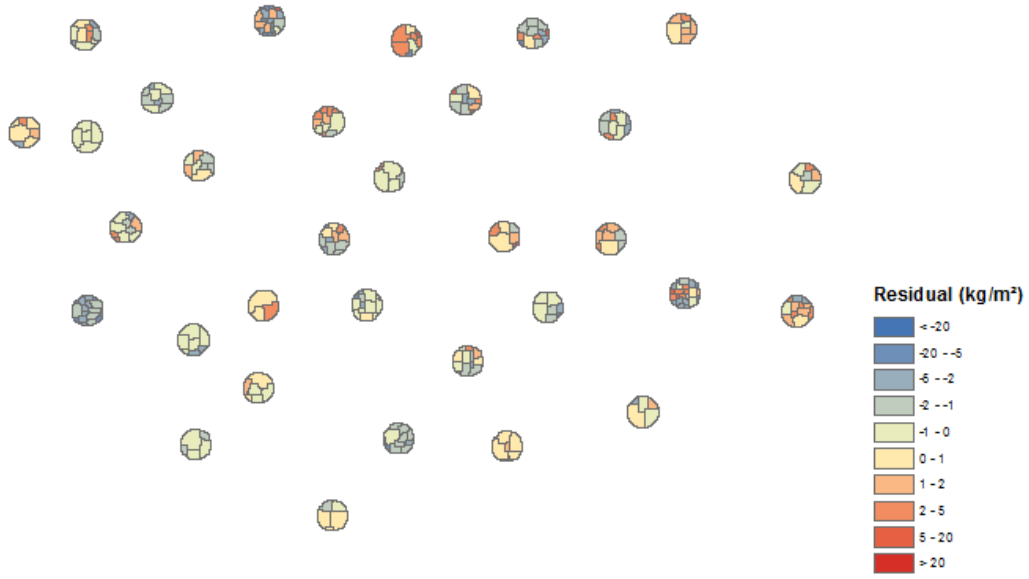


Figure 13 SVM biomass regression residuals at TR. Regression model was built on the height metric layers generated using all neighborhood settings. Data partitioning was not performed.

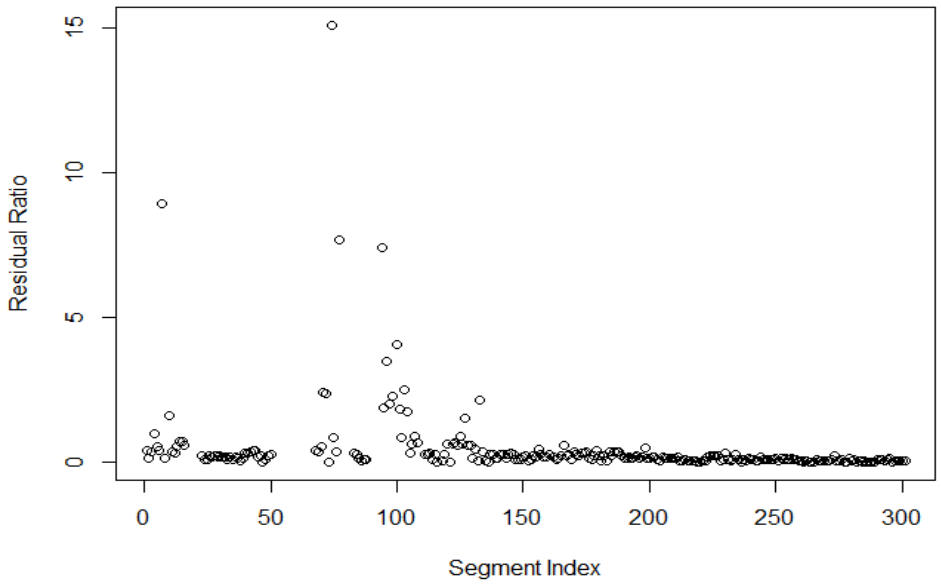


Figure 14 Residual ratio for the selected model at Trinity River site.

5. Conclusions

Random Forest performed equally well on training data in both mountainous (Ozark National Forest) and bottomland (Trinity River) hardwood forests, though its validation accuracy varied over different neighborhood settings. The algorithm suffered from severe over-learning problem. Support vector machines (SVM) showed significant sensitivity to neighborhood settings, generalization ability, and good prediction ability. It produced the most accurate results in both study sites. It is problematic to use a global uniform neighborhood when environmental factors, biophysical characteristics and data quality vary. The ideal neighborhood for a specific scene should be able to capture main features of the vertical distribution of tree components. For the Ozark National Forest study site, a 15 m neighborhood was most successful while all image layers generated using all five neighborhood sizes should be used for the Trinity River study site. The optimal neighborhoods and learning algorithms in generated accurate estimates in both ONF

and TR. Training accuracies equal to 0.988 and 0.990 and validation accuracies equal to 0.902 and 0.853 respectively in the two study sites.

The neighborhood technique for transforming lidar points to images may suffer from the modifiable areal unit problem (MAUP). Large neighborhoods are more likely to produce better results at the cost of losing spatial details. More lidar points outside sampling areas are included in calculation as neighborhood grows in size. The additional information may generate severe biased results as discussed in section 4.1 and 4.2.

Furthermore, lidar data at both study sites were collected in leaf-on season allowing for good modelling of forest tree structures which is a function of tree species, growth cycle, tree healthiness and forest age. Such conditions determine the quality of lidar classification and related image products (e.g. extracting ground points and generating DEMs, finding vegetation returns and constructing canopy height models, identifying homogeneous forest units) which further determine the accuracy and applicability of regression models. Since high-accuracy models have been successfully built for ONF and TR, the forests at both study sites are suitable for segment-level biomass estimation from lidar data.

Future work includes (1) integrate lidar data with high-resolution imagery for individual-tree level estimation and compare the results with segment-level estimations; (2) estimate above ground biomass for the entire study site with parallel computing at both level; (3) examine and describe the vertical and horizontal distribution of tree structures at both study sites.

References

ASPRS. (2005, May 7). ASPRS LAS 1.1 Format Standard. Retrieved from http://www.asprs.org/a/society/committees/standards/asprs_las_format_v11.pdf

Axelsson, P. (1999). Processing of laser scanner data—algorithms and applications. *ISPRS Journal of Photogrammetry and Remote Sensing*, *54*, 138–147.

Blaschke, T. (2010). Object based image analysis for remote sensing. *ISPRS Journal of Photogrammetry and Remote Sensing*, *65*(1), 2–16. doi:10.1016/j.isprsjprs.2009.06.004

Bortolot, Z. J., & Wynne, R. H. (2005). Estimating forest biomass using small footprint LiDAR data: An individual tree-based approach that incorporates training data. *ISPRS Journal of Photogrammetry & Remote Sensing*, *59*(6), 342–360.

Brandtberg, T., Warner, T. A., Landenberger, R. E., & McGraw, J. B. (2003). Detection and analysis of individual leaf-off tree crowns in small footprint, high sampling density lidar data from the eastern deciduous forest in North America. *Remote Sensing of Environment*, *85*(3), 290.

Breiman, L., & Cutler, A. (n.d.). Random Forests. Retrieved from http://www.stat.berkeley.edu/~breiman/RandomForests/cc_home.htm#overview

Brown, S. (1997). Estimating biomass and biomass change of tropical forests : a primer / by Sandra Brown. In *FAO forestry paper ; 134*. Rome : Food and Agriculture Organization of the United Nations, 1997. Retrieved from http://0-search.ebscohost.com.library.uark.edu/login.aspx?direct=true&db=agr&AN=CAT10830302&site=ehost-live&scope=site

Chávez, J. D. y, & Tullis, J. A. (2013). Deciduous Forest Structure Estimated with LIDAR-Optimized Spectral Remote Sensing. *Remote Sensing*, *5*(1), 155–182.

Chen, Q., Gong, P., Baldocchi, D., & Xie GengXin. (2007). Filtering airborne laser scanning data with morphological methods. *PE&RS, Photogrammetric Engineering & Remote Sensing*, *73*(2), 175–185.

Clark, M. L., Roberts, D. A., Ewel, J. J., & Clark, D. B. (2011). Estimation of tropical rain forest aboveground biomass with small-footprint lidar and hyperspectral sensors. *Remote Sensing of Environment*, *115*(11), 2931–2942.

Clinton, N., Holt, A., Scarborough, J., Yan, L., & Gong, P. (2010). Accuracy Assessment Measures for Object-based Image Segmentation Goodness. *Photogrammetric Engineering & Remote Sensing*, *76*(3), 289–299. doi:10.14358/PERS.76.3.289

Colin., Ying, Yiming., Campbell. (2011). Learning with support vector machines. Retrieved from http://dx.doi.org/10.2200/S00324ED1V01Y201102AIM010

Dengsheng Lu. (2006). The potential and challenge of remote sensing-based biomass estimation. *International Journal of Remote Sensing*, *27*(7), 1297–1328.

Determination of optimal scale parameters for alliance-level forest classification of multispectral IKONOS images. (n.d.). Retrieved from http://www.isprs.org/proceedings/xxxvi/4-c42/papers/OBIA2006_Kim_Madden.pdf

Drăguţ, L., Csillik, O., Eisank, C., & Tiede, D. (2014). Automated parameterisation for multi-scale image segmentation on multiple layers. *ISPRS Journal of Photogrammetry and Remote Sensing*, *88*(0), 119–127. doi:10.1016/j.isprsjprs.2013.11.018

Drăguţ, L., Tiede, D., & Levick, S. R. (2010). ESP: a tool to estimate scale parameter for multiresolution image segmentation of remotely sensed data. *International Journal of Geographical Information Science*, *24*(6), 859–871. doi:10.1080/13658810903174803

Drake, J. B., Dubayah, R. O., Clark, D. B., Knox, R. G., Blair, J. B., Hofton, M. A., Prince, S. D. (2002). Estimation of tropical forest structural characteristics using large-footprint lidar. *Remote Sensing of Environment*, *79*(2/3), 305.

Drake, J. B., Knox, R. G., Dubayah, R. O., Clark, D. B., Condit, R., Blair, J. B., & Hofton, M. (2003). Above-ground biomass estimation in closed canopy Neotropical forests using lidar remote sensing: factors affecting the generality of relationships. *Global Ecology & Biogeography*, *12*(2), 147–159.

E.J. Huising, & L.M. Gomes Pereira. (1998). Errors and accuracy estimates of laser data acquired by various laser scanning systems for topographic applications. *ISPRS Journal of Photogrammetry and Remote Sensing*, *53*, 245–261.

Farid, A., Goodrich, D. C., Bryant, R., & Sorooshian, S. (2008). Using airborne lidar to predict Leaf Area Index in cottonwood trees and refine riparian water-use estimates. *Journal of Arid Environments*, *72*(1), 1–15. doi:10.1016/j.jaridenv.2007.04.010

Filippi, A. M., Güneralp, İ., & Randall, J. (2014). Hyperspectral remote sensing of aboveground biomass on a river meander bend using multivariate adaptive regression splines and stochastic gradient boosting. *Remote Sensing Letters*,*5*(5), 432-441.

Forests and Carbon Sotrage. (n.d.). Retrieved from http://www.fs.fed.us/ccrc/topics/forests-carbon

Gleason, C. J., & Im, J. (2012). Forest biomass estimation from airborne LiDAR data using machine learning approaches. *Remote Sensing of Environment*, *125*, 80–91.

Gleason, C. J., & Im JungHo. (2012). A fusion approach for tree crown delineation from lidar data. *PE&RS, Photogrammetric Engineering & Remote Sensing*, *78*(7), 679–692.

Güneralp, İ., Filippi, A. M., & Randall, J. (2014). Estimation of floodplain aboveground biomass using multispectral remote sensing and nonparametric modeling. *International Journal of*

Applied Earth Observation and Geoinformation, *33*(0), 119–126. http://doi.org/10.1016/j.jag.2014.05.004

Haralick, R. M., Sternberg, S. R., & Zhuang, X. (1987). Image analysis using mathematical morphology. *IEEE Transactions On Pattern Analysis And Machine Intelligence*, *9*(4), 532–550.

Harding, D., Lefsky, M., Parker, G., & Blair, J.(2001). Laser altimeter canopy height profiles: methods and validation for closed-canopy, broadleaf forests. *Remote Sensing of Environment*, *76*(3), 283–297. doi:10.1016/S0034-4257(00)00210-8

Huang, M., & Lai, C. (2014, July). Parallelizing computer vision algorithms on acceleration technologies: A SIFT case study. In *Signal and Information Processing (ChinaSIP), 2014 IEEE China Summit & International Conference on* (pp. 325-329). IEEE.

Jenkins, J. C., Birdsey, R. A., Heath, L. S., & Chojnacky, D. C. (2003). National-scale biomass estimators for United States tree species. *Forest Science*, *49*(1), 12–35.

Kato, A., Moskal, L. M., Schiess, P., Swanson, M. E., Calhoun, D., & Stuetzle, W. (2009). Capturing tree crown formation through implicit surface reconstruction using airborne lidar data. *Remote Sensing of Environment*, *113*(6), 1148–1162. doi:10.1016/j.rse.2009.02.010

Keqi Zhang, Shu-ching Chen, Whitman, D., Mei-Ling Shyu, Jianhua Yan, & Chengcui Zhang. (2003). A Progressive Morphological Filter for Removing Nonground Measurements From Airborne LIDAR Data. *IEEE Transactions on Geoscience & Remote Sensing*, *41*(4), 872.

Kronseder, K., Ballhorn, U., Böhm, V., & Siegert, F. (2012). Above ground biomass estimation across forest types at different degradation levels in Central Kalimantan using LiDAR data. *International Journal of Applied Earth Observation and Geoinformation*, *18*(0), 37–48. doi:10.1016/j.jag.2012.01.010

Kuhn, M., Weston, S., Keefer, C., & Coulter, N. (2012, May 11). Cubist Models For Regression. Retrieved from http://cran.rproject.org/web/packages/Cubist/vignettes/cubist.pdf

Lai, C., Huang, M., Shi, X., & You, H. (2013, November). Accelerating geospatial applications on hybrid architectures. In *High Performance Computing and Communications & 2013 IEEE International Conference on Embedded and Ubiquitous Computing (HPCC_EUC), 2013 IEEE 10th International Conference on* (pp. 1545-1552). IEEE.

Lee, H. S., & Youman, N. H. (2003). DTM Extraction of Lidar Returns Via Adaptive Processing. *IEEE Transactions on Geoscience & Remote Sensing*, *41*(9), 2063–2069.

Lefsky, M. A., Cohen, W. B., Harding, D. J., Parker, G. G., Acker, S. A., & Gower, S. T. (2002). Lidar remote sensing of above-ground biomass in three biomes. *Global Ecology & Biogeography*, *11*(5), 393–399.

Lefsky, M. A., Cohen, W. B., Parker, G. G., & Harding, D. J. (2002). Lidar Remote Sensing for Ecosystem Studies. *BioScience*, *52*(1), 19.

Lefsky, M. A., Cohen, W. B., & Spies, T. A. (2001). An evaluation of alternate remote sensing products for forest inventory, monitoring, and mapping of Douglas-fir forests in western Oregon. *Canadian Journal of Forest Research*, *31*(1), 78.

Lefsky, M. A., Parker, G., Shugart, H. H., Harding, D., & Cohen, W. B. (1999). Surface Lidar Remote Sensing of Basal Area and Biomass in Deciduous Forests of Eastern Maryland, USA [electronic resource]. *Remote Sensing of Environment*, *67*(1), 83–98.

Lefsky, M. A., Spies, T. A., Harding, D., Parker, G. G., Cohen, W. B., & Acker, S. A. (1999). Lidar Remote Sensing of the Canopy Structure and Biophysical Properties of Douglas-Fir Western Hemlock Forests [electronic resource]. *Remote Sensing of Environment*, *70*(3), 339–361.

Liaw, A., & Wiener, M. (2012). Classification and Regression by randomForest. *R News*, *2*(3), 18–21.

Lim, K. S., & Treitz, P. M. (2004). Estimation of above ground forest biomass from airborne discrete return laser scanner data using canopy-based quantile estimators. *Scandinavian Journal of Forest Research*, *19*(6), 558–570.

Lovell, J. L., Jupp, D. L. B., Culvenor, D. S., & Coops, N. C. (2003). Using airborne and ground-based ranging lidar to measure canopy structure in Australian forests. *Canadian Journal of Remote Sensing*, *29*(5), 607–622. doi:10.5589/m03-026

Lu, D. S., Chen, Q., Wang, G. X., Moran, E., Batistella, M., Zhang MaoZhen, Saah, D. (2012a). Aboveground forest biomass estimation with landsat and LiDAR data and uncertainty analysis of the estimates. *International Journal of Forestry Research*, *2012*, Article ID 436537–Article ID 436537.

Lu, D. S., Chen, Q., Wang, G. X., Moran, E., Batistella, M., Zhang MaoZhen, Saah, D. (2012b). Aboveground forest biomass estimation with landsat and LiDAR data and uncertainty analysis of the estimates. *International Journal of Forestry Research*, *2012*, Article ID 436537–Article ID 436537.

Magnussen, S., & Boudewyn, P. (1998). Derivations of stand heights from airborne laser scanner data with canopy-based quantile estimators. *Canadian Journal of Forest Research*, *28*(7), 1016–1031.

Martin Baatz, & Arno Schape. (n.d.). Multiresolution Segmentation: an optimization approach for high quality multi-scale image segmentation. Retrieved from http://www.ecognition.cc/download/baatz_schaepe.pdf

Maynard, J. J., & Johnson, M. G. (2014). Scale-dependency of LiDAR derived terrain attributes in quantitative soil-landscape modeling: Effects of grid resolution vs. neighborhood extent. *Geoderma*, *230–231*(0), 29–40. doi:10.1016/j.geoderma.2014.03.021

Means, J. E., Acker, S. A., Fitt, B. J., Renslow, M., Emerson, L., & Hendrix, C. J. (2000). Predicting forest stand characteristics with airborne scanning lidar. *PE&RS, Photogrammetric Engineering & Remote Sensing*, *66*(11), 1367–1371.

Meyer, D. (2014, January 10). Support Vector Machines (The Interface to libsvm in package e1071). Retrieved from http://cran.r-project.org/web/packages/e1071/vignettes/svmdoc.pdf

Multiresolution segmentation: a parallel approach for high resolution image segmentation in multicore architectures. (n.d.). Retrieved from http://www.isprs.org/proceedings/xxxviii/4-c7/pdf/Happ_143.pdf

Muss, J. D., Mladenoff, D. J., & Townsend, P. A. (2011). A pseudo-waveform technique to assess forest structure using discrete lidar data. *Remote Sensing of Environment*, *115*(3), 824–835.

Næsset, E. (2002). Predicting forest stand characteristics with airborne scanning laser using a practical two-stage procedure and field data. *Remote Sensing of Environment*, *80*(1), 88–99. doi:10.1016/S0034-4257(01)00290-5

Popescu, S. C. (2007). Estimating biomass of individual pine trees using airborne lidar. *Biomass & Bioenergy*, *31*(9), 646–655.

Popescu, S. C., Scrivani, J. A., & Wynne, R. H. (2004). Fusion of small-footprint lidar and multispectral data to estimate plot-level volume and biomass in deciduous and pine forests in Virginia, USA. *Forest Science*, *50*(4), 551–565.

Popescu, S. C., Wynne, R. H., & Nelson, R. F. (2003). Measuring individual tree crown diameter with lidar and assessing its influence on estimating forest volume and biomass. *Canadian Journal of Remote Sensing*, *29*(5), 564–577. doi:10.5589/m03-027

Popescu, S. C., & Zhao, K. (2008). A voxel-based lidar method for estimating crown base height for deciduous and pine trees. *Remote Sensing of Environment*, *112*(3), 767–781. doi:10.1016/j.rse.2007.06.011

Quinlan, J. R. (1986). Induction of decision trees. *Machine Learning*, *1*(1), 81–106. doi:10.1007/BF00116251

Quinlan, J. R. (n.d.-a). *C4.5 : programs for machine learning*.

Quinlan, J. R. (n.d.-b). *Combining Instance-Based and Model-Based Learning*. Retrieved from http://cs.ecs.baylor.edu/~hamerly/courses/5325_11s/papers/ibl/quinlan1993combining.pdf

Quinlan, J. R. (n.d.-c). *Learning with continuous classes*. Retrieved from http://sci2s.ugr.es/keel/pdf/algorithm/congreso/1992-Quinlan-AI.pdf

Riggins, J. J., Tullis, J. A., & Stephen, F. M. (2009). Per-segment aboveground forest biomass estimation using LIDAR-derived height percentile statistics. *GIScience and Remote Sensing*, *46*(2), 232–248.

Scott, C. T., & Gove, J. H. (2002). Forest inventory. Encyclopedia of Environmetrics.

Shan, J., & Toth, C. K. (2009). *Topographic Laser Ranging and Scanning: Principles and Processing*. Boca Raton: CRC Press/Taylor & Francis Group.

Tab, F. A., Naghdy, G., & Mertins, A. (2006). Scalable multiresolution color image segmentation. *Signal Processing*, *86*(7), 1670–1687. doi:10.1016/j.sigpro.2005.09.016

Unnikrishnan, R., Pantofaru, C., & Hebert, M. (2007). Toward objective evaluation of image segmentation algorithms. *IEEE Transactions On Pattern Analysis And Machine Intelligence*, *29*(6), 929–944.

Vaglio Laurin, G., Chen, Q., Lindsell, J. A., Coomes, D. A., Frate, F. D., Guerriero, L., Valentini, R. (2014). Above ground biomass estimation in an African tropical forest with lidar and hyperspectral data. *ISPRS Journal of Photogrammetry and Remote Sensing*, *89*(0), 49–58. doi:10.1016/j.isprsjprs.2014.01.001

Van Aardt, J. A. N., Oderwald, R. G., & Wynne, R. H. (2006). Forest Volume and Biomass Estimation Using Small-Footprint Lidar-Distributional Parameters on a Per-Segment Basis. *Forest Science*, *52*(6), 636–649.

Wei-Yin Loh. (n.d.). Classification and regression trees. Retrieved from http://www.stat.wisc.edu/~loh/treeprogs/guide/wires11.pdf

Woodcock, C. E., & Strahler, A. H. (1987). The factor of scale in remote sensing. *Remote Sensing of Environment*, *21*(3), 311–332.

Yang, J., Li, P., & He, Y. (2014). A multi-band approach to unsupervised scale parameter selection for multi-scale image segmentation. *ISPRS Journal of Photogrammetry and Remote Sensing*, *94*(0), 13–24. doi:10.1016/j.isprsjprs.2014.04.008

Yang, W., Ni-Meister, W., & Lee, S. (2011). Assessment of the impacts of surface topography, off-nadir pointing and vegetation structure on vegetation lidar waveforms using an extended geometric optical and radiative transfer model. *Remote Sensing of Environment*, *115*(11), 2810–2822.

Zhang, H., Fritts, J. E., & Goldman, S. A. (2008). Image segmentation evaluation: A survey of unsupervised methods. *Computer Vision and Image Understanding*, *110*(2), 260–280. doi:10.1016/j.cviu.2007.08.003

Zhou, X., & Hemstrom, M. A. (2009). Estimating aboveground tree biomass on forest land in the Pacific Northwest : a comparison of approaches / Xiaoping Zhou and Miles A. Hemstrom. In *Research paper PNW ; RP-584*. Portland, OR : U.S. Dept. of Agriculture, Forest Service,

Pacific Northwest Research Station, [2009]. Retrieved from http://www.fs.fed.us/pnw/pubs/pnw_rp584.pdf

Zolkos, S. G., Goetz, S. J., & Dubayah, R. (2013). A meta-analysis of terrestrial aboveground biomass estimation using lidar remote sensing. *Remote Sensing of Environment*, *128*(0), 289–298. doi:10.1016/j.rse.2012.10.017

Appendix

Appendix 1 Model evaluation statistics at ONF							
ONF GS1NR0.5							
SVM	Fold	RMSE_tr	COR_tr	R^2-adj_tr	RMSE_va	COR_va	R^2-adj_va
	1	718.021	0.502	0.248	1354.578	0.275	0.024
	2	87.047	0.998	0.996	821.025	0.327	0.051
	3	82.381	0.998	0.995	994.562	0.382	0.096
	4	85.736	0.998	0.995	759.509	0.086	-0.055
	5	87.345	0.998	0.996	629.862	0.177	-0.023
	6	88.550	0.998	0.995	433.442	-0.042	-0.061
	7	721.523	0.582	0.334	1220.800	0.103	-0.051
	8	88.635	0.998	0.996	431.800	0.320	0.046
	9	782.113	0.431	0.181	1198.602	0.203	-0.015
	10	87.820	0.998	0.996	661.011	-0.014	-0.062
	Ave	282.917	0.850	0.773	850.519	0.182	-0.005
	All	84.844	0.998	0.995	NA	NA	NA
Cubist	Fold	RMSE_tr	COR_tr	R^2-adj_tr	RMSE_va	COR_va	R^2-adj_va
	1	799.196	0.261	0.062	1392.989	0.238	0.004
	2	864.289	0.340	0.110	571.024	0.408	0.114
	3	720.119	0.641	0.408	1083.745	0.183	-0.023
	4	888.100	0.265	0.065	783.761	0.120	-0.047
	5	881.704	0.338	0.109	621.888	0.289	0.033
	6	889.762	0.337	0.108	285.551	0.259	0.009
	7	687.433	0.665	0.439	1240.075	0.051	-0.060
	8	916.982	0.261	0.062	332.742	0.301	0.034
	9	829.250	0.306	0.088	1170.584	0.018	-0.058
	10	893.730	0.363	0.127	280.712	0.608	0.330
	Ave	837.056	0.378	0.158	776.307	0.247	0.033
	All	841.058	0.346	0.115	NA	NA	NA
Random Forest	Fold	RMSE_tr	COR_tr	R^2-adj_tr	RMSE_va	COR_va	R^2-adj_va
	1	366.309	0.943	0.888	447.669	0.211	-0.009
	2	431.061	0.934	0.871	175.467	0.544	0.252
	3	389.702	0.941	0.884	379.417	0.083	-0.052
	4	404.266	0.938	0.878	268.515	0.112	-0.049
	5	414.283	0.940	0.883	190.409	0.493	0.201
	6	427.121	0.938	0.880	153.658	0.216	-0.013
	7	373.658	0.941	0.885	400.471	-0.038	-0.061

	Fold	RMSE_tr	COR_tr	R²-adj_tr	RMSE_va	COR_va	R²-adj_va
	8	409.687	0.940	0.883	166.620	0.318	0.045
	9	397.675	0.933	0.870	367.978	0.189	-0.021
	10	428.211	0.935	0.874	166.881	0.299	0.032
	Ave	404.197	0.938	0.880	271.709	0.243	0.033
	All	401.952	0.942	0.887	NA	NA	NA
ONF GS1NR2.5							
SVM	Fold	RMSE_tr	COR_tr	R²-adj_tr	RMSE_va	COR_va	R²-adj_va
	1	1628.023	0.717	0.508	3774.756	0.811	0.609
	2	1943.484	0.639	0.402	3474.669	0.211	-0.092
	3	1993.310	0.740	0.542	547.865	0.827	0.648
	4	192.358	0.997	0.994	4426.811	0.080	-0.104
	5	2042.383	0.707	0.494	795.170	0.938	0.863
	6	1823.913	0.796	0.629	1303.317	0.675	0.388
	7	1786.343	0.801	0.638	1225.120	0.080	-0.136
	8	1810.709	0.794	0.626	458.008	0.780	0.552
	9	1844.993	0.791	0.621	434.117	0.967	0.925
	10	1995.792	0.727	0.522	743.355	0.441	0.105
	Ave	1706.131	0.771	0.598	1718.319	0.581	0.376
	All	1782.580	0.779	0.603	NA	NA	NA
Cubist	Fold	RMSE_tr	COR_tr	R²-adj_tr	RMSE_va	COR_va	R²-adj_va
	1	1493.410	0.836	0.696	4630.330	0.047	-0.140
	2	1038.761	0.915	0.835	3487.812	0.282	-0.052
	3	1886.335	0.872	0.757	697.989	0.708	0.445
	4	717.653	0.972	0.944	4191.790	0.351	0.025
	5	2025.396	0.786	0.614	887.214	0.950	0.888
	6	1621.985	0.886	0.782	1222.696	0.734	0.481
	7	1993.246	0.793	0.625	967.547	0.146	-0.119
	8	1925.188	0.860	0.737	264.411	0.948	0.883
	9	2292.560	0.593	0.344	1267.965	0.748	0.497
	10	2288.410	0.623	0.381	1076.443	-0.063	-0.107
	Ave	1728.294	0.814	0.671	1869.420	0.485	0.280
	All	1608.084	0.881	0.773	NA	NA	NA
Random Forest	Fold	RMSE_tr	COR_tr	R²-adj_tr	RMSE_va	COR_va	R²-adj_va
	1	886.853	0.941	0.885	1307.411	0.504	0.147
	2	980.390	0.945	0.892	937.572	0.627	0.307
	3	1033.374	0.949	0.900	313.630	0.768	0.544
	4	740.673	0.966	0.932	1423.515	0.550	0.225

	Fold	RMSE_tr	COR_tr	R²-adj_tr	RMSE_va	COR_va	R²-adj_va
	5	1074.653	0.947	0.896	208.182	0.958	0.906
	6	1082.132	0.941	0.885	526.369	0.624	0.313
	7	1069.676	0.944	0.890	290.615	0.139	-0.121
	8	1061.902	0.946	0.894	96.245	0.902	0.786
	9	1067.338	0.942	0.886	125.693	0.985	0.966
	10	1064.937	0.944	0.890	493.519	0.139	-0.090
	Ave	1006.193	0.947	0.895	572.275	0.619	0.398
	All	1025.503	0.944	0.891	NA	NA	NA
ONF GS1NR5							
SVM	Fold	RMSE_tr	COR_tr	R²-adj_tr	RMSE_va	COR_va	R²-adj_va
	1	1470.508	0.493	0.235	697.981	0.570	0.241
	2	1555.906	0.398	0.149	747.180	0.501	0.168
	3	1313.338	0.647	0.413	954.576	0.646	0.353
	4	1496.597	0.465	0.208	309.539	0.892	0.770
	5	1450.823	0.494	0.236	850.741	0.636	0.330
	6	416.999	0.873	0.760	4365.447	0.182	-0.074
	7	1503.021	0.469	0.212	389.834	0.877	0.743
	8	1358.897	0.607	0.361	827.009	0.749	0.517
	9	1510.620	0.473	0.215	238.418	0.927	0.846
	10	1280.847	0.678	0.454	967.921	0.022	-0.124
	Ave	1335.755	0.560	0.324	1034.865	0.600	0.377
	All	1430.011	0.477	0.221	NA	NA	NA
Cubist	Fold	RMSE_tr	COR_tr	R²-adj_tr	RMSE_va	COR_va	R²-adj_va
	1	959.980	0.929	0.862	508.721	0.811	0.614
	2	911.994	0.921	0.846	788.235	0.453	0.117
	3	289.020	0.985	0.970	629.426	0.765	0.539
	4	1126.767	0.894	0.798	292.280	0.912	0.811
	5	1531.646	0.421	0.169	985.384	0.488	0.143
	6	536.987	0.765	0.581	4348.421	0.225	-0.055
	7	1059.418	0.887	0.784	998.460	0.286	-0.021
	8	1035.244	0.925	0.855	823.692	0.605	0.302
	9	1590.263	0.396	0.148	480.241	0.524	0.202
	10	829.244	0.942	0.886	2249.165	-0.080	-0.118
	Ave	987.056	0.807	0.690	1210.402	0.499	0.254
	Fold	RMSE_tr	COR_tr	R²-adj_tr	RMSE_va	COR_va	R²-adj_va
	All	393.051	0.970	0.940	NA	NA	NA

Random Forest	Fold	RMSE_tr	COR_tr	R^2-adj_tr	RMSE_va	COR_va	R^2-adj_va
	1	847.656	0.908	0.822	157.725	0.824	0.638
	2	834.637	0.911	0.827	291.391	0.508	0.175
	3	842.547	0.909	0.825	257.065	0.679	0.401
	4	849.747	0.905	0.818	152.732	0.820	0.631
	5	823.113	0.911	0.828	250.709	0.746	0.501
	6	256.661	0.958	0.917	1481.779	0.149	-0.087
	7	814.533	0.915	0.836	227.504	0.612	0.306
	8	836.851	0.907	0.821	219.464	0.749	0.517
	9	838.127	0.912	0.830	198.097	0.624	0.329
	10	802.058	0.929	0.861	583.596	-0.074	-0.119
	Ave	774.593	0.916	0.839	382.006	0.564	0.329
	All	792.326	0.914	0.834	NA	NA	NA
ONF GS1NR10							
SVM	Fold	RMSE_tr	COR_tr	R^2-adj_tr	RMSE_va	COR_va	R^2-adj_va
	1	292.495	0.981	0.961	577.538	0.891	0.773
	2	300.318	0.980	0.959	352.522	0.972	0.940
	3	249.703	0.984	0.968	808.744	0.886	0.763
	4	374.587	0.965	0.930	783.787	0.826	0.651
	5	352.593	0.965	0.930	1835.763	0.569	0.257
	6	115.915	0.996	0.993	613.400	0.943	0.878
	7	274.291	0.984	0.968	555.787	0.863	0.715
	8	281.949	0.980	0.960	817.832	0.783	0.575
	9	292.845	0.981	0.962	139.369	0.992	0.982
	10	298.066	0.980	0.961	269.474	0.925	0.841
	Ave	283.276	0.980	0.959	675.422	0.865	0.738
	All	286.005	0.981	0.962	NA	NA	NA
Cubist	Fold	RMSE_tr	COR_tr	R^2-adj_tr	RMSE_va	COR_va	R^2-adj_va
	1	471.939	0.962	0.925	777.390	0.833	0.664
	2	519.194	0.952	0.906	477.666	0.938	0.867
	3	505.210	0.943	0.888	1069.834	0.738	0.498
	4	426.522	0.961	0.923	1292.473	0.432	0.105
	5	496.769	0.928	0.861	1959.823	0.302	0.000
	6	617.859	0.912	0.830	1111.823	0.750	0.514
	7	538.473	0.939	0.881	516.979	0.813	0.623
	8	637.904	0.927	0.858	473.056	0.926	0.843
	9	637.176	0.917	0.840	307.166	0.970	0.934
	10	599.511	0.915	0.835	716.924	0.608	0.307

	Fold	RMSE_tr	COR_tr	R^2-adj_tr	RMSE_va	COR_va	R^2-adj_va
	Ave	545.056	0.936	0.875	870.313	0.731	0.536
	All	536.587	0.942	0.886	NA	NA	NA
Random Forest	Fold	RMSE_tr	COR_tr	R^2-adj_tr	RMSE_va	COR_va	R^2-adj_va
	1	443.572	0.963	0.928	311.933	0.712	0.458
	2	427.502	0.965	0.930	190.229	0.894	0.780
	3	411.658	0.965	0.930	347.178	0.756	0.528
	4	423.116	0.966	0.932	356.834	0.592	0.286
	5	348.157	0.971	0.942	596.380	0.502	0.178
	6	446.997	0.957	0.915	335.026	0.780	0.564
	7	431.579	0.964	0.929	279.949	0.594	0.281
	8	408.941	0.966	0.932	281.425	0.768	0.548
	9	457.214	0.960	0.921	126.224	0.933	0.857
	10	437.080	0.961	0.923	300.647	0.379	0.058
	Ave	423.582	0.964	0.928	312.582	0.691	0.454
	All	419.336	0.966	0.933	NA	NA	NA
ONF GS1NR15							
SVM	Fold	RMSE_tr	COR_tr	R^2-adj_tr	RMSE_va	COR_va	R^2-adj_va
	1	166.143	0.996	0.992	675.561	0.966	0.923
	2	164.143	0.996	0.992	395.538	0.984	0.963
	3	169.196	0.996	0.992	175.087	0.988	0.973
	4	164.133	0.996	0.992	343.470	0.958	0.903
	5	176.232	0.995	0.991	596.017	0.947	0.883
	6	174.656	0.995	0.991	499.661	0.970	0.930
	7	160.957	0.995	0.990	1705.714	0.902	0.783
	8	390.493	0.977	0.953	1026.375	0.961	0.911
	9	162.850	0.996	0.993	421.809	0.933	0.850
	10	159.271	0.997	0.993	329.466	0.956	0.900
	Ave	188.807	0.994	0.988	616.870	0.957	0.902
	All	161.350	0.996	0.992	NA	NA	NA
Cubist	Fold	RMSE_tr	COR_tr	R^2-adj_tr	RMSE_va	COR_va	R^2-adj_va
	1	527.088	0.962	0.924	1009.447	0.952	0.890
	2	578.738	0.957	0.915	649.886	0.956	0.899
	3	671.727	0.935	0.872	489.166	0.898	0.775
	4	521.191	0.969	0.938	302.451	0.967	0.924
	5	430.776	0.975	0.949	858.078	0.909	0.802
	6	571.480	0.961	0.921	510.867	0.958	0.905
	7	502.434	0.955	0.911	2339.397	0.713	0.427

	Fold	RMSE_tr	COR_tr	R²-adj_tr	RMSE_va	COR_va	R²-adj_va
	8	575.701	0.949	0.898	1647.075	0.783	0.549
	9	514.897	0.966	0.932	867.094	0.790	0.562
	10	565.806	0.960	0.921	324.779	0.950	0.887
	Ave	545.984	0.959	0.918	899.824	0.888	0.762
	Fold	RMSE_tr	COR_tr	R²-adj_tr	RMSE_va	COR_va	R²-adj_va
	All	479.871	0.968	0.937	NA	NA	NA
Random Forest	Fold	RMSE_tr	COR_tr	R²-adj_tr	RMSE_va	COR_va	R²-adj_va
	1	487.879	0.971	0.943	248.613	0.942	0.868
	2	496.859	0.967	0.935	178.625	0.971	0.933
	3	498.658	0.970	0.940	139.306	0.934	0.852
	4	499.442	0.970	0.941	212.054	0.844	0.664
	5	446.967	0.976	0.952	411.637	0.823	0.632
	6	498.857	0.970	0.940	199.781	0.962	0.914
	7	413.701	0.970	0.940	713.738	0.835	0.647
	8	432.875	0.975	0.949	578.980	0.747	0.484
	9	498.491	0.970	0.941	66.540	0.984	0.962
	10	514.759	0.968	0.937	145.727	0.921	0.823
	Ave	478.849	0.971	0.942	289.500	0.896	0.778
	All	474.500	0.972	0.944	NA	NA	NA
ONF GS1NRa							
SVM	Fold	RMSE_tr	COR_tr	R²-adj_tr	RMSE_va	COR_va	R²-adj_va
	1	823.305	0.884	0.780	433.093	0.905	0.805
	2	1000.801	0.793	0.625	370.387	0.768	0.555
	3	971.938	0.825	0.677	362.474	0.740	0.514
	4	963.376	0.813	0.658	763.688	0.336	0.039
	5	796.180	0.895	0.800	457.358	0.649	0.373
	6	951.121	0.826	0.680	639.280	0.426	0.123
	7	1019.603	0.774	0.596	807.863	0.936	0.865
	8	810.238	0.889	0.789	674.379	0.910	0.815
	9	1017.371	0.787	0.617	267.192	0.825	0.655
	10	75.203	0.997	0.993	3809.621	0.259	-0.018
	Ave	842.914	0.848	0.722	858.533	0.675	0.472
	All	783.5869	0.88314	0.778373	NA	NA	NA
Cubist	Fold	RMSE_tr	COR_tr	R²-adj_tr	RMSE_va	COR_va	R²-adj_va
	1	1454.151	0.317	0.093	926.733	0.394	0.090
	2	1233.531	0.762	0.578	438.310	0.593	0.297
	3	1200.035	0.742	0.548	264.697	0.841	0.685

	Fold	RMSE_tr	COR_tr	R²-adj_tr	RMSE_va	COR_va	R²-adj_va
	4	1171.181	0.791	0.623	495.631	0.470	0.156
	5	1265.924	0.558	0.305	368.319	0.820	0.644
	6	1376.874	0.562	0.310	277.754	0.753	0.536
	7	1393.234	0.345	0.112	1256.628	0.813	0.634
	8	806.786	0.968	0.937	1107.265	0.480	0.166
	9	1303.145	0.590	0.343	274.662	0.710	0.463
	10	208.561	0.988	0.976	3880.870	0.220	-0.038
	Ave	1141.342	0.662	0.483	929.087	0.609	0.363
	All	922.730	0.917	0.840	NA	NA	NA
Random Forest	Fold	RMSE_tr	COR_tr	R²-adj_tr	RMSE_va	COR_va	R²-adj_va
	1	696.977	0.968	0.937	263.279	0.658	0.389
	2	693.559	0.961	0.922	145.190	0.763	0.547
	3	694.651	0.966	0.933	229.337	0.390	0.087
	4	658.722	0.969	0.939	194.645	0.612	0.322
	5	693.196	0.961	0.922	219.745	0.388	0.080
	6	666.920	0.967	0.934	157.052	0.729	0.498
	7	637.821	0.967	0.934	450.318	0.414	0.102
	8	710.564	0.956	0.913	388.996	0.098	-0.073
	9	680.992	0.964	0.928	145.484	0.738	0.507
	10	305.005	0.962	0.924	1264.968	0.215	-0.040
	Ave	643.841	0.964	0.929	345.901	0.500	0.242
	All	611.927	0.967	0.934	NA	NA	NA

Appendix 2 Model evaluation statistics at TR							
TR GS1NR0.5							
SVM	Fold	RMSE_tr	COR_tr	R²-adj_tr	RMSE_va	COR_va	R²-adj_va
	1	281.464	0.816	0.664	314.267	0.680	0.450
	2	267.475	0.820	0.671	337.934	0.715	0.499
	3	245.700	0.854	0.729	487.919	0.578	0.318
	4	274.648	0.820	0.671	286.418	0.872	0.754
	5	281.410	0.799	0.638	286.556	0.804	0.638
	6	261.103	0.816	0.664	382.403	0.761	0.569
	7	293.053	0.789	0.621	345.438	0.632	0.385
	8	278.428	0.808	0.651	299.338	0.770	0.583
	9	276.274	0.807	0.651	351.591	0.741	0.539
	10	289.288	0.794	0.630	280.205	0.805	0.639
	Ave	274.884	0.812	0.659	337.207	0.736	0.537

	Fold	RMSE_tr	COR_tr	R^2-adj_tr	RMSE_va	COR_va	R^2-adj_va
	All	289.539	0.788	0.621	NA	NA	NA
Cubist	Fold	RMSE_tr	COR_tr	R^2-adj_tr	RMSE_va	COR_va	R^2-adj_va
	1	226.161	0.897	0.805	257.871	0.754	0.559
	2	219.482	0.893	0.796	292.646	0.793	0.620
	3	215.317	0.903	0.816	360.706	0.633	0.386
	4	216.997	0.910	0.828	270.561	0.860	0.733
	5	197.957	0.919	0.843	294.176	0.784	0.606
	6	206.799	0.906	0.820	358.103	0.809	0.647
	7	202.010	0.915	0.837	324.759	0.687	0.459
	8	210.561	0.906	0.821	335.438	0.694	0.469
	9	200.782	0.917	0.841	352.300	0.733	0.526
	10	225.813	0.892	0.795	254.075	0.828	0.678
	Ave	212.188	0.906	0.820	310.063	0.758	0.568
	All	231.001	0.886	0.785	NA	NA	NA
Random Forest	Fold	RMSE_tr	COR_tr	R^2-adj_tr	RMSE_va	COR_va	R^2-adj_va
	1	147.284	0.957	0.915	345.376	0.668	0.434
	2	145.105	0.957	0.915	334.107	0.708	0.490
	3	132.685	0.965	0.930	347.024	0.671	0.437
	4	141.993	0.958	0.918	266.696	0.832	0.685
	5	137.379	0.961	0.924	258.888	0.840	0.699
	6	144.080	0.955	0.912	372.161	0.777	0.594
	7	134.320	0.962	0.926	340.998	0.622	0.372
	8	145.459	0.956	0.914	336.516	0.667	0.431
	9	132.260	0.963	0.928	355.040	0.709	0.491
	10	136.303	0.961	0.924	281.514	0.781	0.600
	Ave	139.687	0.960	0.921	323.832	0.728	0.523
	All	148.394	0.954	0.910	NA	NA	NA
TR GS1NR2.5							
SVM	Fold	RMSE_tr	COR_tr	R^2-adj_tr	RMSE_va	COR_va	R^2-adj_va
	1	202.150	0.952	0.907	360.292	0.858	0.727
	2	143.146	0.979	0.958	325.070	0.882	0.770
	3	213.169	0.947	0.896	356.258	0.976	0.951
	4	200.181	0.956	0.915	326.227	0.716	0.496
	5	251.176	0.932	0.868	394.242	0.810	0.644
	6	259.434	0.924	0.853	261.920	0.919	0.840
	7	217.082	0.944	0.891	268.262	0.956	0.910
	8	224.508	0.935	0.873	570.359	0.804	0.633

	Fold	RMSE_tr	COR_tr	R^2-adj_tr	RMSE_va	COR_va	R^2-adj_va
	9	174.477	0.966	0.933	498.540	0.681	0.444
	10	208.625	0.952	0.906	303.731	0.891	0.788
	Ave	209.395	0.949	0.900	366.490	0.849	0.720
	All	211.775	0.948	0.899	NA	NA	NA
Cubist	Fold	RMSE_tr	COR_tr	R^2-adj_tr	RMSE_va	COR_va	R^2-adj_va
	1	176.790	0.965	0.930	351.131	0.868	0.744
	2	190.279	0.960	0.921	216.077	0.962	0.922
	3	176.955	0.965	0.931	349.701	0.957	0.913
	4	189.480	0.963	0.927	323.099	0.706	0.480
	5	167.574	0.970	0.942	393.959	0.825	0.669
	6	175.487	0.966	0.933	254.197	0.923	0.847
	7	188.489	0.959	0.920	304.385	0.909	0.821
	8	177.255	0.962	0.926	467.829	0.877	0.760
	9	173.771	0.966	0.933	272.175	0.932	0.864
	10	162.115	0.973	0.946	268.412	0.909	0.821
	Ave	177.820	0.965	0.931	320.096	0.887	0.784
	All	178.659	0.964	0.930	NA	NA	NA
Random Forest	Fold	RMSE_tr	COR_tr	R^2-adj_tr	RMSE_va	COR_va	R^2-adj_va
	1	150.160	0.976	0.953	348.877	0.869	0.746
	2	155.273	0.975	0.951	281.863	0.924	0.848
	3	152.695	0.975	0.951	341.871	0.943	0.885
	4	150.858	0.977	0.955	310.449	0.738	0.529
	5	140.295	0.980	0.961	395.492	0.835	0.687
	6	156.653	0.975	0.950	305.769	0.892	0.789
	7	139.217	0.980	0.959	344.511	0.896	0.795
	8	140.142	0.977	0.954	515.667	0.836	0.687
	9	155.659	0.975	0.950	279.475	0.907	0.817
	10	153.722	0.977	0.954	292.025	0.896	0.797
	Ave	149.467	0.977	0.954	341.600	0.874	0.758
	All	145.783	0.978	0.956	NA	NA	NA
TR GS1NR5							
SVM	Fold	RMSE_tr	COR_tr	R^2-adj_tr	RMSE_va	COR_va	R^2-adj_va
	1	459.500	0.815	0.662	263.482	0.869	0.746
	2	429.056	0.817	0.667	659.145	0.777	0.589
	3	439.523	0.825	0.679	521.390	0.727	0.508
	4	425.594	0.836	0.698	541.449	0.549	0.274
	5	371.452	0.876	0.766	506.194	0.590	0.321

	Fold	RMSE_tr	COR_tr	R²-adj_tr	RMSE_va	COR_va	R²-adj_va
	6	367.922	0.876	0.767	515.283	0.688	0.453
	7	453.513	0.810	0.654	412.500	0.766	0.570
	8	356.711	0.879	0.772	585.288	0.644	0.390
	9	435.030	0.817	0.666	669.340	0.688	0.451
	10	294.183	0.909	0.826	904.068	0.538	0.260
	Ave	403.248	0.846	0.716	557.814	0.684	0.456
	All	369.578	0.871	0.757	NA	NA	NA
Cubist	Fold	RMSE_tr	COR_tr	R²-adj_tr	RMSE_va	COR_va	R²-adj_va
	1	261.610	0.957	0.915	500.669	0.690	0.456
	2	294.802	0.941	0.884	710.191	0.700	0.470
	3	290.977	0.942	0.886	555.751	0.673	0.430
	4	294.198	0.941	0.886	513.063	0.605	0.341
	5	313.694	0.932	0.868	424.567	0.681	0.441
	6	308.260	0.931	0.865	594.666	0.424	0.147
	7	295.008	0.943	0.889	382.177	0.804	0.632
	8	302.311	0.932	0.867	776.661	0.445	0.163
	9	294.098	0.939	0.881	606.425	0.768	0.573
	10	298.301	0.926	0.856	895.020	0.548	0.271
	Ave	295.326	0.938	0.880	595.919	0.634	0.392
	All	283.225	0.946	0.895	NA	NA	NA
Random Forest	Fold	RMSE_tr	COR_tr	R²-adj_tr	RMSE_va	COR_va	R²-adj_va
	1	282.725	0.943	0.888	393.852	0.779	0.591
	2	259.992	0.946	0.894	616.910	0.819	0.658
	3	264.820	0.947	0.897	526.944	0.711	0.485
	4	277.005	0.944	0.891	482.294	0.651	0.401
	5	271.387	0.946	0.895	439.736	0.648	0.396
	6	247.310	0.955	0.912	597.702	0.493	0.213
	7	256.716	0.952	0.905	379.459	0.818	0.656
	8	232.156	0.959	0.919	749.069	0.464	0.181
	9	251.364	0.950	0.902	630.347	0.699	0.467
	10	208.902	0.965	0.931	912.525	0.527	0.248
	Ave	255.238	0.951	0.903	572.884	0.661	0.430
	All	251.355	0.952	0.907	NA	NA	NA
TR GS1NR10							
SVM	Fold	RMSE_tr	COR_tr	R²-adj_tr	RMSE_va	COR_va	R²-adj_va
	1	269.199	0.950	0.902	1369.654	0.538	0.253
	2	476.854	0.874	0.762	630.701	0.753	0.542

	Fold	RMSE_tr	COR_tr	R^2-adj_tr	RMSE_va	COR_va	R^2-adj_va
	3	467.446	0.877	0.769	430.584	0.907	0.814
	4	522.403	0.847	0.716	288.711	0.955	0.907
	5	522.471	0.850	0.722	291.136	0.935	0.867
	6	515.251	0.856	0.732	269.279	0.934	0.865
	7	511.906	0.852	0.725	420.864	0.885	0.772
	8	525.293	0.852	0.724	256.589	0.917	0.832
	9	524.371	0.840	0.704	462.764	0.897	0.793
	10	512.823	0.859	0.737	214.513	0.957	0.912
	Ave	484.802	0.866	0.749	463.479	0.868	0.756
	All	495.679	0.861	0.740	NA	NA	NA
Cubist	Fold	RMSE_tr	COR_tr	R^2-adj_tr	RMSE_va	COR_va	R^2-adj_va
	1	223.055	0.970	0.941	1417.953	0.493	0.206
	2	439.744	0.900	0.810	707.468	0.695	0.454
	3	404.819	0.919	0.844	561.918	0.840	0.691
	4	353.328	0.942	0.886	277.282	0.969	0.936
	5	439.920	0.908	0.823	297.362	0.927	0.853
	6	427.220	0.913	0.833	217.180	0.958	0.913
	7	417.771	0.914	0.835	343.344	0.931	0.859
	8	427.733	0.913	0.833	243.624	0.930	0.858
	9	361.629	0.927	0.858	745.642	0.718	0.488
	10	390.012	0.930	0.863	325.377	0.887	0.775
	Ave	388.523	0.924	0.853	513.715	0.835	0.703
	All	342.631	0.944	0.891	NA	NA	NA
Random Forest	Fold	RMSE_tr	COR_tr	R^2-adj_tr	RMSE_va	COR_va	R^2-adj_va
	1	185.459	0.978	0.957	1387.545	0.512	0.225
	2	272.655	0.965	0.931	660.810	0.731	0.508
	3	291.011	0.960	0.921	378.828	0.931	0.860
	4	305.442	0.957	0.915	254.651	0.968	0.934
	5	281.341	0.964	0.930	254.896	0.961	0.920
	6	291.813	0.961	0.924	327.274	0.903	0.805
	7	274.467	0.966	0.932	438.519	0.873	0.750
	8	299.313	0.959	0.920	266.936	0.923	0.843
	9	288.236	0.960	0.921	580.394	0.829	0.670
	10	277.971	0.965	0.931	387.633	0.864	0.733
	Ave	276.771	0.964	0.928	493.749	0.849	0.725
	All	277.914	0.963	0.927	NA	NA	NA

TR GS1NR15							
SVM	Fold	RMSE_tr	COR_tr	R^2-adj_tr	RMSE_va	COR_va	R^2-adj_va
	1	398.095	0.905	0.819	396.460	0.935	0.867
	2	354.982	0.923	0.851	481.278	0.886	0.775
	3	411.640	0.900	0.809	276.354	0.977	0.953
	4	106.950	0.995	0.989	310.622	0.925	0.847
	5	104.294	0.995	0.989	553.916	0.950	0.898
	6	311.609	0.948	0.898	562.770	0.673	0.426
	7	90.465	0.996	0.992	611.520	0.807	0.632
	8	276.638	0.954	0.910	544.251	0.856	0.719
	9	323.580	0.939	0.882	688.899	0.630	0.365
	10	105.434	0.994	0.988	469.368	0.949	0.895
	Ave	248.369	0.955	0.913	489.544	0.859	0.738
	All	104.334	0.995	0.989	NA	NA	NA
Cubist	Fold	RMSE_tr	COR_tr	R^2-adj_tr	RMSE_va	COR_va	R^2-adj_va
	1	248.219	0.969	0.938	490.963	0.856	0.718
	2	221.579	0.972	0.945	428.243	0.912	0.824
	3	241.949	0.967	0.935	164.777	0.979	0.956
	4	230.415	0.973	0.946	342.929	0.901	0.801
	5	201.363	0.977	0.955	522.955	0.887	0.776
	6	222.331	0.975	0.950	299.790	0.925	0.848
	7	220.613	0.973	0.947	407.046	0.919	0.836
	8	223.614	0.973	0.946	471.536	0.912	0.823
	9	242.578	0.967	0.935	468.350	0.827	0.668
	10	219.241	0.973	0.947	578.172	0.870	0.745
	Ave	227.190	0.972	0.944	417.476	0.899	0.799
	All	228.756	0.971	0.943	NA	NA	NA
Random Forest	Fold	RMSE_tr	COR_tr	R^2-adj_tr	RMSE_va	COR_va	R^2-adj_va
	1	183.110	0.982	0.965	473.776	0.905	0.809
	2	190.356	0.980	0.960	472.125	0.891	0.784
	3	192.825	0.981	0.962	130.099	0.988	0.974
	4	192.763	0.981	0.962	383.727	0.875	0.752
	5	177.977	0.984	0.968	460.715	0.912	0.822
	6	191.560	0.981	0.962	353.578	0.881	0.765
	7	184.124	0.981	0.963	409.550	0.920	0.837
	8	164.954	0.985	0.970	510.409	0.877	0.757
	9	186.592	0.982	0.964	441.511	0.847	0.703
	10	191.784	0.980	0.959	506.231	0.926	0.851

	Fold	RMSE_tr	COR_tr	R^2-adj_tr	RMSE_va	COR_va	R^2-adj_va
	Ave	185.605	0.982	0.964	414.172	0.902	0.805
	All	178.069	0.983	0.967	NA	NA	NA
TR GS1NRa							
SVM	Fold	RMSE_tr	COR_tr	R^2-adj_tr	RMSE_va	COR_va	R^2-adj_va
	1	69.314	0.994	0.988	357.651	0.882	0.770
	2	63.306	0.995	0.991	247.497	0.902	0.807
	3	62.298	0.996	0.991	201.020	0.930	0.860
	4	62.510	0.995	0.991	215.170	0.898	0.799
	5	62.073	0.995	0.991	207.915	0.934	0.868
	6	62.777	0.995	0.990	220.556	0.953	0.906
	7	61.238	0.995	0.991	293.203	0.901	0.805
	8	63.226	0.995	0.990	177.792	0.958	0.915
	9	70.955	0.994	0.988	154.872	0.950	0.900
	10	60.424	0.995	0.989	358.549	0.949	0.897
	Ave	63.812	0.995	0.990	243.422	0.926	0.853
	All	62.156	0.995	0.991	NA	NA	NA
Cubist	Fold	RMSE_tr	COR_tr	R^2-adj_tr	RMSE_va	COR_va	R^2-adj_va
	1	107.268	0.987	0.975	272.225	0.942	0.884
	2	94.277	0.990	0.980	223.715	0.870	0.748
	3	89.832	0.990	0.981	249.735	0.895	0.793
	4	77.619	0.994	0.988	153.324	0.946	0.892
	5	78.564	0.993	0.985	182.560	0.949	0.898
	6	90.894	0.992	0.984	195.573	0.957	0.914
	7	102.523	0.989	0.978	197.736	0.954	0.907
	8	69.959	0.994	0.988	173.309	0.967	0.932
	9	80.595	0.992	0.985	177.151	0.934	0.869
	10	70.358	0.993	0.986	418.532	0.929	0.857
	Ave	86.189	0.991	0.983	224.386	0.934	0.869
	All	68.729	0.995	0.989	NA	NA	NA
Random Forest	Fold	RMSE_tr	COR_tr	R^2-adj_tr	RMSE_va	COR_va	R^2-adj_va
	1	102.918	0.987	0.975	269.400	0.953	0.905
	2	100.999	0.989	0.979	273.128	0.864	0.738
	3	108.547	0.987	0.973	249.466	0.890	0.784
	4	104.669	0.988	0.976	197.491	0.910	0.823
	5	103.215	0.988	0.977	322.700	0.830	0.677
	6	107.055	0.987	0.975	228.104	0.941	0.882
	7	106.392	0.987	0.974	225.980	0.941	0.882

	Fold	RMSE_tr	COR_tr	R^2-adj_tr	RMSE_va	COR_va	R^2-adj_va
	8	105.192	0.988	0.976	182.873	0.969	0.937
	9	107.515	0.988	0.976	186.812	0.928	0.855
	10	107.198	0.985	0.970	365.752	0.956	0.910
	Ave	105.370	0.987	0.975	250.171	0.918	0.839
	All	100.902	0.989	0.977	NA	NA	NA